Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

**Комплексна бакалаврська кваліфікаційна робота на тему:**

# «ІНФОРМАЦІЙНА СИСТЕМА «РОЗУМНИЙ БУДИНОК» НА БАЗІ ЛОКАЛЬНОЇ КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ». АНАЛІЗ ДАНИХ ТА ВИБІР ОПТИМАЛЬНОЇ АРХІТЕКТУРИ СИСТЕМИ

Виконав: студент 4 курсу, групи 2ІСТ-21б спеціальності 126 – «Інформаційні системи та технології»
_____ Сергій КРИШТАЛЬ
Керівник: к.т.н., доц. каф. САІТ
_____ Дмитро ПРОЦЕНКО
«05» 06 2025 р.
Рецензент: к.ф.-м.н., доц. каф. КН
_____ Олександр ШЕВЧУК
«10» 06 2025 р.

**Допущено до захисту**

Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«06» 06 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 126 – «Інформаційні системи та технології»
Освітньо-професійна програма – Прикладні інформаційні технології

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
"28" 03 2025 року

**ЗАВДАННЯ**
**НА КОМПЛЕКСНУ БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ**
**СТУДЕНТУ**
Кришталю Сергію Олександровичу

1. Тема роботи: «Інформаційна система «розумний будинок» на базі локальної клієнт-серверної архітектури». Аналіз даних та вибір оптимальної архітектури системи
керівник роботи: Проценко Д. П., к.т.н., доц. каф. САІТ
затверджені наказом ВНТУ від «20» 03 2025 року №97
2. Термін подання студентом роботи 31.05.25
3. Вихідні дані до роботи:
1) Дані про мережу пристроїв «розумний будинок».
2) Дані про користувачів та сценарії використання системи.
4. Зміст текстової частини:
1) Аналіз сучасних систем «розумний будинок» і архітектурних рішень.
2) Вибір оптимальної клієнт-серверної архітектури з локальною обробкою подій.
3) Розроблення інформаційної системи для управління «розумним будинком».
5. Перелік ілюстративного матеріалу:
1) Компонентна діаграма архітектури інформаційної системи;
2) Діаграма розгортання із взаємодією пристроїв;
3) Діаграма USE CASE інформаційної системи;.
4) Діаграма послідовності взаємодії сенсора, MQTT-брокера, Home Assistant та виконавчого пристрою;
5) Приклад графічного інтерфейсу користувача Home Assistant.

6. Консультанти розділів роботи:

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання 28.03.25 р.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва та зміст етапу | Термін виконання | | Примітк |
|---|---|---|---|---|
| | | початок | закінчення | |
| 1 | Аналіз предметної області | 01.04 | 15.04 | вик |
| 2 | Порівняльний аналіз проблематики | 15.04 | 30.04 | вик |
| 3 | Вибір оптимальних інформаційних технологій | 01.05 | 10.05 | вик |
| 4 | Розроблення інформаційної системи автоматизації роботи користувачів | 10.05 | 20.05 | вик |
| 5 | Оформлення матеріалів до захисту БКР | 20.05 | 30.05 | вик |

Студент Сергій КРИШТА

Керівник роботи Дмитро ПРОЦЕН

# АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 64 сторінок формату А4, на яких є 13 рисунків, список використаних джерел містить 20 найменувань.

Метою роботи є розроблення інформаційної системи «розумний будинок», аналіз даних та вибір оптимальної архітектури.

В роботі обрано архітектуру інформаційної системи, створено діаграми взаємодії, послідовності та автоматизацій, а також представлено приклад програмної реалізації із використанням YAML-конфігурацій для Home Assistant, також наведено приклади реалізації аналізу даних в інформаційній системі

Ключові слова: розумний будинок, інформаційна система, Home Assistant, MQTT, Zigbee, клієнт-сервер, локальна архітектура, автоматизація.

# ABSTRACT

The bachelor's qualification thesis consists of 64 A4 pages and includes 13 figures. The list of references contains 20 sources.

The aim of the thesis is to choise a smart home information system, perform data analysis, and select the optimal system architecture.

The work presents the design of the information system architecture, the creation of interaction, sequence, and automation diagrams, as well as an example of software implementation using YAML configurations for Home Assistant. Additionally, examples of data analysis implementation within the system are provided.



Keywords**:** smart home, information system, Home Assistant, MQTT, Zigbee, client-server, local architecture, automation.

**ЗМІСТ**

# ВСТУП

**Актуальність теми**. Сучасний світ все більше орієнтується на автоматизацію та інтелектуальні технології, особливо в сфері житла. Системи «розумного будинку» стають не просто зручним рішенням, а необхідністю для ефективного управління енергоресурсами, підвищення безпеки та комфорту. Зростання попиту на такі системи пов’язане зі стрімким розвитком IoT-пристроїв, доступністю та зміною способу життя людей. За даними досліджень, у 2020 році кількість підключених “розумних” пристроїв у домогосподарствах перевищила 1 мільярд одиниць, а до кінця 2025 року очікується зростання до 1,5 мільярдів пристроїв. Прогнозують, що до 2030 року понад 45% усіх домогосподарств у розвинених країнах будуть використовувати ті чи інші системи домашньої автоматизації. Ці тенденції підтверджують актуальність розробки нових, більш досконалих систем домашньої автоматизації, особливо таких, що поєднують в собі надійність локальних рішень з сучасними можливостями інтелектуального управління.

**Мета і задачі дослідження.** є вибір оптимальної архітектури з метою обґрунтування для забезпечення її ефективності, надійності та масштабованості, і подальшого апаратного розширення системи за допомогою аналізу даних.

Для досягнення поставленої мети необхідно розв’язати такі задачі:

– Провести аналіз сучасних тенденцій у сфері «розумних будинків» та існуючих систем автоматизації;

– Визначити оптимальні технології для реалізації системи (мови програмування, архітектура, модулі);

– Обрати платформу для реалізації інформаційної системи, на локальної клієнт-серверної архітектури

– Оцінити вплив різних варіантів архітектур (однорівневої, дворівневої, трирівневої та багаторівневої) на ефективність, стабільність та масштабованість функціонування системи.

**Об’єктом дослідження** є інформаційна система «Розумний будинок» та її функціонування в умовах локальної клієнт-серверної архітектури.

**Предметом дослідження** є процес обґрунтованого вибору оптимальної архітектури інформаційної системи «Розумний будинок» на базі локальної клієнт-серверної моделі з урахуванням технічних, функціональних та експлуатаційних вимог.

**Публікації.** За результатами даної роботи була зроблена доповідь на тему «Аналіз даних та вибір оптимальної архітектури в інформаційній системі «розумний будинок» на базі Home Assistant» на Міжнародній науково-практичній інтернет-конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2025) з публікацією тез [1].

Робота виконувалась на замовлення кафедри САІТ Вінницького національного технічного університету.

# 1 АНАЛІЗ ПРОБЛЕМАТИКИ СИСТЕМ «РОЗУМНИЙ БУДИНОК»

## 1.1 Аналіз проблематики у сфері «розумних будинків»

Системи “розумного будинку” представляють собою складні інформаційно-технічні комплекси, покликані підвищити рівень комфорту, енергоефективності, безпеки та автоматизації в житловому середовищі. Однак, незважаючи на стрімкий розвиток технологій Інтернету речей (IoT), бездротових систем, існує низка ключових проблем, що стримують повноцінну інтеграцію таких рішень у повсякденне життя.

Однією з найбільш актуальних проблем є відсутність єдиного стандарту взаємодії між пристроями. На ринку існує значна кількість рішень від різних виробників, які використовують власні протоколи зв’язку (Zigbee, Z-Wave, Bluetooth LE, Thread Matter тощо) (табл. 1.1), що ускладнює інтеграцію і потребує проміжного програмного або апаратного забезпечення. Це веде до фрагментації систем і підвищує вартість впровадження.

Таблиця 1.1 – Порівняльні характеристики протоколів бездротової передачі даних у системах розумного будинку

| Протокол | Радіус дії | Швидкість | Енергоспоживання | Взаємодія |
|---|---|---|---|---|
| Zigbee | Середній | Середня | Низьке | Висока |
| Wi-Fi | Високий | Висока | Високе | Низька |
| Bluetooth LE | Низький | Низька | Низьке | Середня |
| Matter | Середній | Середня | Низьке | Висока |

Інша ключова проблема — безпека та конфіденційність даних (рис. 1.1). В умовах зростаючої кількості кібератак навіть побутові пристрої можуть стати вразливими точками входу до домашньої мережі, створюючи ризики несанкціонованого доступу до особистої інформації, втрати контролю над системою або її критичної дестабілізації.

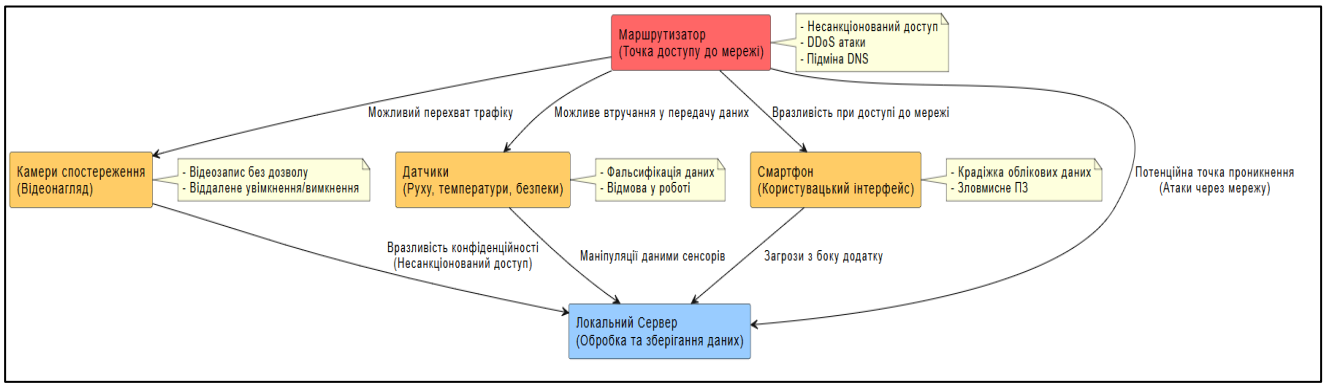


Рисунок 1.1 – Основні точки потенційних кіберзагроз у системі розумного будинку

Багато пристроїв не мають достатнього рівня захисту (шифрування даних, автентифікація, регулярні оновлення), що створює ризики для особистих даних користувачів, контролю над пристроями або навіть фізичної безпеки.

Крім того, питання енергоспоживання й автономності залишаються невирішеними для багатьох пристроїв. З одного боку, автономні сенсори потребують мінімального енергоспоживання або альтернативних джерел живлення. З іншого боку, централізовані вузли (шлюзи, сервери) мають працювати безперебійно, а отже — потребують резервного живлення та стабільної мережі.

Також варто відзначити низьку адаптивність до наявної інфраструктури в старих житлових будинках. Встановлення сенсорів, електропроводки, маршрутизаторів і шлюзів без порушення інтер’єру або капітального ремонту часто є складним завданням. Це обмежує впровадження систем у масовому сегменті житла.

Ще одним важливим аспектом є складність проєктування та налаштування систем для пересічного користувача. Багато сучасних рішень вимагають технічної обізнаності, знання мережевих технологій або налаштувань мікроконтролерів. Через це користувачі часто відмовляються від розширених функцій на користь базових рішень, що знижує потенційну ефективність систем.

З точки зору економіки, висока початкова вартість обладнання та монтажу стримує широке впровадження систем “розумного будинку”, особливо в країнах з невисоким рівнем доходів. Хоча з часом така система може зекономити ресурсу

(електроенергію, воду, газ), її вартість не завжди швидко окупається, особливо без державної підтримки або субсидування.

Соціально-психологічним бар'єром є низький рівень цифрової грамотності серед населення, особливо старшого покоління. Багато користувачів не довіряють системам автоматизації або вважають їх надмірно складними. Крім того, виникають побоювання щодо постійного спостереження, втрати контролю або залежності від технологій.

## 1.2 Технології автоматизації житлових приміщень

Сучасні тенденції розвитку інформаційних технологій сприяють активному впровадженню систем автоматизації у житловій інфраструктурі. Автоматизоване керування інженерними мережами житлових приміщень дозволяє не лише підвищити рівень комфорту мешканців, а й забезпечити оптимальне використання енергоресурсів, підвищити рівень безпеки та знизити витрати на експлуатацію будівель. Основу подібних систем становлять апаратні засоби — сенсори, виконавчі механізми та контролери, — а також програмне забезпечення для централізованого або розподіленого керування (рис. 1.2).

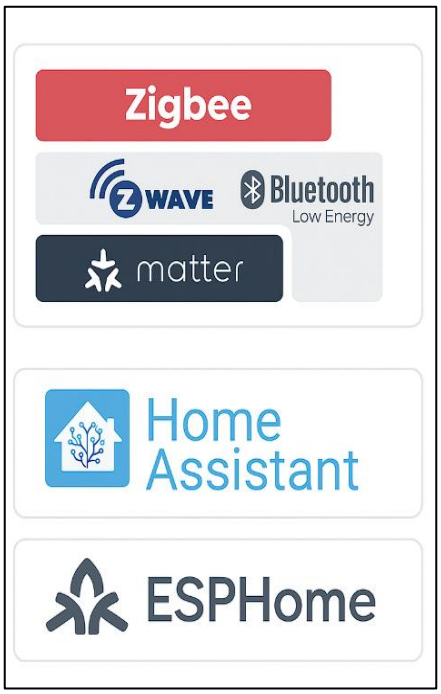


Рисунок 1.2 - Технології та компоненти автоматизації розумного будинку

Одним із ключових елементів таких систем є технології бездротової передачі даних. Їх застосування значно спрощує процес інтеграції пристроїв у вже існуюче середовище, уникаючи потреби в прокладанні кабельних мереж. Найбільш поширеними серед них є Zigbee, Z-Wave, Bluetooth Low Energy (BLE), Wi-Fi, Thread та новітній протокол Matter ( табл. 1.2).

Zigbee є одним із найдавніших і найстабільніших стандартів для створення сіткових мереж сенсорів і виконавчих пристроїв. Він працює у стандарті IEEE 802.15.4, забезпечуючи низьке енергоспоживання, високу надійність з'єднання та підтримку топології типу mesh, що дозволяє збільшити дальність передачі сигналу завдяки ретрансляції між вузлами. Його перевагою є широка підтримка виробниками обладнання та висока сумісність із популярними платформами розумного будинку[2].

Z-Wave — ще один спеціалізований протокол, орієнтований на домашню автоматизацію. Він працює в частотних діапазонах, не зайнятих Wi-Fi чи Bluetooth-пристроями, що зменшує ймовірність перешкод. Крім того, Z-Wave відомий своєю простотою конфігурації, стабільністю і зворотною сумісністю між різними поколіннями пристроїв. Обидва протоколи (Zigbee і Z-Wave) потребують використання спеціальних шлюзів або хабів для взаємодії з центральними контролерами.

Bluetooth Low Energy (BLE) використовується переважно для пристроїв з обмеженими енергетичними ресурсами, наприклад, сенсорів руху чи температури. Його основна перевага — мінімальне енергоспоживання при достатньо високій швидкості обміну даними на короткі відстані. Проте BLE менш ефективний у створенні масштабованих мереж, тому зазвичай застосовується як допоміжна технологія.

Wi-Fi, попри свою популярність у побутовому використанні, не завжди є оптимальним рішенням для автоматизації, оскільки характеризується високим енергоспоживанням. Проте Wi-Fi використовується для пристроїв, що потребують високої пропускної здатності або постійного підключення до локальної мережі, зокрема камер відеоспостереження, медіа-серверів або голосових помічників.

З-поміж новітніх технологій особливу увагу привертає Matter — відкритий прикладний протокол, який підтримується більшістю провідних компаній у сфері IoT (Apple, Google, Amazon, Samsung та інші). Matter дозволяє створювати уніфіковані мережі «розумного будинку», де пристрої різних виробників можуть взаємодіяти без потреби у проміжному програмному забезпеченні або шлюзах.

Таблиця 1.2 - Порівняльна характеристика бездротових технологій для автоматизації житлових приміщень.

| Характеристика | Zigbee | Z-Wave | Wi-Fi | Bluetooth LE | Matter |
|---|---|---|---|---|---|
| Діапазон частот | 2.4 ГГц | 868/915 МГц | 2.4 / 5 ГГц | 2.4 ГГц | IP (Wi-Fi, Ethernet, Thread) |
| Топологія мережі | Mesh | Mesh | Зірка | Зірка / P2P | Mesh / IP |
| Радіус дії (приміщення) | ~10–30 м | ~30–100 м | ~30–50 м | ~5–10 м | Залежить від носія |
| Енергоспоживання | Низьке | Дуже низьке | Високе | Дуже низьке | Залежить від носія |
| Пропускна здатність | Низька | Низька | Висока | Низька | Середня |
| Сумісність між пристроями | Обмежена | Висока (але вузький ринок) | Висока | Висока | Висока (стандартизована) |
| Необхідність шлюзу | Так | Так | Ні | Ні | Ні (опціонально) |
| Призначення | Сенсори, лампи, реле | Сенсори, замки, реле | Камери, хаби, медіа | Трекери, сенсори | Універсальне рішення |

Основна перевага Matter — підтримка IP-мереж, що дає змогу інтегрувати систему в існуючу інфраструктуру без модифікацій. Важливою технічною складовою сучасних систем автоматизації є використання мікроконтролерних платформ. Зокрема, мікроконтролери ESP32[15], завдяки вбудованим модулям Wi-Fi і Bluetooth, невисокій вартості, багатофункціональності та широкій підтримці з боку розробницьких спільнот, стали популярним вибором для реалізації локальних автоматизованих вузлів. У поєднанні з прошивкою ESPHome ці мікроконтролери дозволяють швидко інтегрувати пристрої в екосистему Home Assistant — одного з найпоширеніших відкритих рішень для реалізації локального керування без потреби в хмарних сервісах.

Програмні платформи, зокрема Home Assistant, OpenHAB, Domoticz, є серцевиною логіки розумного будинку. Вони дозволяють здійснювати централізоване управління всіма підключеними пристроями, формувати сценарії автоматизації, аналізувати дані сенсорів та інтегрувати різні протоколи в єдину інформаційну модель. Особливість таких систем полягає в підтримці локального виконання без підключення до Інтернету, що забезпечує високий рівень приватності, швидкості та автономності.

Таким чином, технології автоматизації житлових приміщень формують комплексну систему, яка поєднує в собі енергоефективні бездротові протоколи, мікроконтролери з відкритою архітектурою та інтероперабельні програмні платформи. Їх грамотне використання дає змогу створити надійні, безпечні та масштабовані рішення, здатні адаптуватися до потреб конкретного користувача або типу житла.

## 1.3 Переваги та недоліки використання системи «розумний будинок»

Переваги використання системи «Розумний дім»:

1. Підвищення енергоефективності. Системи «розумного дому» значно знижують споживання електроенергії за рахунок автоматичного, адаптивного управління різними побутовими пристроями. Наприклад, інтелектуальні термостати збирають дані про температуру повітря, вологість, присутність

людей у кімнатах та погодні умови за вікном. Вони автоматично регулюють роботу опалення або кондиціонера, знижуючи температуру, коли мешканці відсутні, і підвищуючи її перед їх поверненням. Також розумні датчики руху та присутності контролюють освітлення — світло автоматично вимикається в кімнатах, де нікого немає, що усуває забуті ввімкнені лампи. Аналогічно, система може вимкнути непотрібні електроприлади або регулювати напругу, що підвищує ефективність використання електроенергії. В довгостроковій перспективі це не лише зменшує загальне споживання енергії, а й дозволяє значно економити на комунальних платежах, а також зменшує навантаження на мережу та екологічний слід будинку.

2. Зручність та комфорт. Одним із головних переваг «розумного дому» є можливість централізованого керування всіма системами житла через один інтерфейс — це може бути смартфон, планшет або навіть настінна панель управління. Замість того, щоб фізично ходити по будинку і вмикати/вимикати світло, закривати штори чи керувати побутовою технікою, користувач робить це дистанційно або автоматично. Сучасні системи також підтримують голосові асистенти (наприклад, Google Assistant, Amazon Alexa, Apple Siri), що дозволяє керувати будинком голосом — дуже зручно, коли руки зайняті або в темряві. Крім того, автоматичні сценарії дозволяють готувати будинок до приходу мешканців, наприклад, підігрівати кімнати взимку перед поверненням додому або відкривати ворота при наближенні автомобіля. Це підвищує рівень комфорту і робить життя більш приємним.

3. Підвищення безпеки. Системи «розумного дому» значно покращують безпеку житла за допомогою сучасних датчиків і камер. Камери відеоспостереження з можливістю трансляції відео в реальному часі дають змогу власнику контролювати будинок будь-де через мобільний додаток. Датчики руху здатні фіксувати підозрілу активність і миттєво повідомляти користувача. Також системи інтегрують детектори диму, чадного газу, витоку води або газу, що дозволяє запобігти аварійним ситуаціям та негайно прийняти заходи. В разі загрози автоматично може активуватися сигналізація, включитися освітлення чи

навіть перекритися подача газу та води. Така інтеграція значно підвищує рівень безпеки мешканців та їхнього майна.

4. Можливість автоматизації рутинних процесів. Одним із ключових аспектів «розумного дому» є можливість створення індивідуальних сценаріїв автоматизації, які спрощують повсякденне життя. Наприклад, можна запрограмувати систему так, щоб при виході з дому автоматично вимикалися всі електроприлади (телевізор, освітлення, праска), зачинялися вікна та двері, активувалася охоронна сигналізація. Це знижує ризик забути вимкнути щось важливе, що може призвести до пожежі чи інших небезпечних ситуацій. Аналогічно можна налаштувати сценарії, які автоматично включають комфортне освітлення ввечері, регулюють штори або запуск побутової техніки у певний час. Завдяки цьому мешканці можуть зосередитися на важливіших справах, довіряючи рутинні дії своїй системі.

5. Підвищення вартості нерухомості. Будинки, оснащені сучасними системами «розумного дому», стають більш привабливими для потенційних покупців і орендарів. Це пов’язано з тим, що такі помешкання демонструють інноваційність, технологічність та піклування власника про комфорт і безпеку. Крім того, автоматизація допомагає знизити експлуатаційні витрати (енергія, опалення, безпека), що підвищує загальну цінність житла. На ринку нерухомості такі будинки можуть мати більш вигідні умови продажу або оренди, а також швидше знаходити покупців. Це інвестиція в майбутнє, яка приносить як економічні, так і комфортні переваги.

Недоліки використання системи «Розумний дім»:

1. Висока вартість встановлення. Повноцінна система «розумного дому» передбачає комплексне обладнання — сервери для обробки даних, різноманітні датчики (руху, освітлення, температури, вологості, газу), контролери для керування приладами, а також автоматичні виконавчі пристрої (реле, двигуни штор тощо). Вартість самого обладнання може бути значною, особливо якщо обирати якісні, надійні пристрої від перевірених виробників. До того ж, необхідно врахувати витрати на проектування системи під конкретні потреби будинку, професійний монтаж кабелів та налаштування всіх

компонентів, що часто потребує залучення кваліфікованих спеціалістів. Хоча з розвитком технологій та зростанням популярності «розумних» рішень ціни поступово падають, на старті інвестиції можуть бути досить вагомими, що може стримувати бажання встановлювати такі системи.

2. Складність обслуговування та налаштування. Інтеграція в єдину систему безлічі різнорідних пристроїв, які можуть працювати на різних протоколах і стандартах, часто створює труднощі для користувача. Регулярне оновлення програмного забезпечення, підтримка сумісності між різними виробниками, усунення технічних несправностей – все це потребує або певних знань, або допомоги фахівця. Для користувача, який не має досвіду у сфері IT чи автоматики, це може стати проблемою — з’являються ризики неправильного налаштування або несправності, що погіршують комфорт і надійність системи. Деякі елементи можуть вимагати регулярної діагностики та заміни батарей, додаткового технічного обслуговування, що додає незручностей.

3. Потенційна загроза конфіденційності. Хоча локальні системи «розумного дому» зазвичай працюють у межах приватної мережі, багато користувачів все ж підключають окремі пристрої до хмарних сервісів для розширення функціоналу (наприклад, камери відеоспостереження, голосові помічники). Це створює потенційні ризики витоку персональних даних, якщо захист цих каналів недостатній або якщо пристрої мають уразливості у безпеці. Хакери можуть отримати доступ до відео, аудіо чи іншої інформації, що стосується приватного життя мешканців. Неправильне налаштування прав доступу або використання слабких паролів також збільшує ризики. Тому забезпечення високого рівня кібербезпеки стає критично важливим аспектом при використанні таких систем.

4. Залежність від електроенергії. Основна частина «розумного дому» працює від електрики, тому у випадку відключення електропостачання більшість функцій стають недоступними. Якщо система не має резервних джерел живлення (UPS або батарей), навіть просте відключення на декілька хвилин може порушити роботу безпеки, опалення, освітлення та інших важливих компонентів. Хоча деякі датчики та пристрої обладнуються вбудованими

акумуляторами або батарейками, їх ресурс обмежений і вони можуть підтримувати роботу системи лише певний час. Для критично важливих функцій (сигналізація, датчики диму) необхідно продумати резервне живлення, що збільшує вартість та складність системи.

5. Ризик морального застарівання. Технології «розумного дому» розвиваються дуже швидко — кожен рік з'являються нові протоколи, пристрої з більшою функціональністю, покращеним інтерфейсом і безпекою. Встановлена система може досить швидко морально застаріти, перестаючи підтримувати нові стандарти або програми. Власнику доводиться або миритися з обмеженим функціоналом, або інвестувати у модернізацію, заміну окремих елементів чи навіть повний апгрейд системи. Це пов'язано з додатковими витратами і зусиллями. Крім того, компоненти системи можуть припинити підтримку з боку виробника (припинити оновлення програмного забезпечення, припинити виробництво запасних частин), що ускладнює подальшу експлуатацію.

## 1.4 Огляд існуючих інформаційних систем для управління «розумними будинками»

У сучасному інформаційному суспільстві, що характеризується стрімким розвитком технологій автоматизації, зростаючими вимогами до енергоефективності, безпеки та комфорту житла, особливої актуальності набувають інформаційні системи, орієнтовані на побудову інтелектуального житлового простору — так званих систем «розумного будинку». Подібні рішення забезпечують централізоване, автоматизоване або напів-автоматизоване керування технічними підсистемами, такими як освітлення, клімат-контроль, системи безпеки, енергоспоживання, побутова техніка, мультимедійні засоби та інші функціональні модулі сучасного житлового середовища.

Загальна архітектура таких систем зазвичай передбачає наявність центрального вузла управління — сервера або хаба, який об'єднує в єдину мережу усі інтегровані пристрої та сенсори. Обмін даними відбувається через спеціалізовані протоколи зв'язку, серед яких найбільш розповсюдженими є

MQTT, Zigbee, Z-Wave, Wi-Fi, Bluetooth LE, Thread, Matter тощо. До критеріїв, які впливають на вибір інформаційної платформи для розгортання системи розумного будинку, належать: тип архітектури (локальна, хмарна або гібридна), рівень відкритості та ліцензування, наявність графічного або текстового інтерфейсу конфігурації, масштабованість, енергоефективність, підтримка пристроїв сторонніх виробників, доступність для кінцевих користувачів тощо.

Однією з найбільш популярних відкритих платформ у цій сфері є Home Assistant [1] — повнофункціональна система з відкритим вихідним кодом, що забезпечує гнучке налаштування автоматизації побутових процесів. Home Assistant орієнтована на локальне розгортання, що дає змогу забезпечити високий рівень конфіденційності даних — усі вимірювання, стани пристроїв та сценарії обробляються безпосередньо на локальному сервері без передачі до хмари. Програма підтримує величезну кількість інтеграцій, що охоплюють популярні бренди, стандарти та протоколи, зокрема MQTT [2], Zigbee, Z-Wave, Wi-Fi, Thread, Matter, Bluetooth LE тощо.

Особливістю Home Assistant є її універсальність у налаштуванні. Користувач має змогу використовувати як візуальний веб-інтерфейс (Lovelace UI) для створення панелей керування та автоматизацій, так і більш гнучкий метод — редагування конфігураційних YAML-файлів [3]. Платформа розповсюджується на умовах вільної ліцензії та може бути розгорнута на різноманітних апаратних засобах: Raspberry Pi, Linux-сервери, міні-ПК, NAS-пристрої, а також віртуальні машини на базі Windows/macOS. Розвинена екосистема, активна спільнота та регулярне оновлення коду роблять Home Assistant потужним інструментом для реалізації як домашніх, так і напівпромислових систем автоматизації [4].

Ще одним гідним прикладом відкритої системи є openHAB (Open Home Automation Bus) — потужна платформа, орієнтована на ентузіастів та професіоналів. Вона базується на мові програмування Java та використовує модульну архітектуру OSGi, яка забезпечує динамічне підключення та оновлення модулів під час роботи системи. OpenHAB також підтримує велику кількість протоколів (Zigbee, Z-Wave, EnOcean, Wi-Fi тощо) і забезпечує

конфігурацію через власну мову сценаріїв DSL або графічний інтерфейс (Paper UI, Basic UI, HABPanel). Завдяки стабільності та широким можливостям інтеграції, openHAB часто застосовується в умовах промислових і корпоративних рішень, а також у великих приватних будинках і смарт-кварталах [5].

Для користувачів, які шукають простішу та менш ресурсомістку альтернативу, доцільним вибором може стати Domoticz. Це легка за вагою система, оптимізована для роботи на пристроях з обмеженими ресурсами, таких як Raspberry Pi Zero. Domoticz підтримує широкий спектр протоколів (Z-Wave, 1-Wire, MQTT, EnOcean, Wi-Fi) та має простий, але інтуїтивно зрозумілий веб-інтерфейс. Хоча функціональні можливості Domoticz дещо поступаються іншим платформам, її стабільність, низьке споживання ресурсів і можливість локального розгортання роблять її придатною для створення бюджетних, але надійних домашніх систем автоматизації.

У сфері закритих комерційних рішень важливе місце займає Apple HomeKit — система, глибоко інтегрована в екосистему пристроїв Apple. Архітектура HomeKit є гібридною: основна обробка команд та сценаріїв здійснюється локально через Apple TV або HomePod, але для дистанційного доступу та резервного копіювання задіяно хмарні сервіси iCloud. Основними комунікаційними протоколами є Bluetooth LE, Wi-Fi, Thread та Matter. З огляду на політику Apple, до HomeKit допускаються лише сертифіковані пристрої (за стандартом MFi — Made for iPhone), що, з одного боку, обмежує кількість сумісних пристроїв, але, з іншого — забезпечує високу стабільність, надійність і безпеку функціонування. Візуальна складова представлена мобільним додатком «Дом» (Home), доступним для iOS/macOS-платформ, з інтеграцією голосового помічника Siri.

Ще одним прикладом гібридного підходу є Samsung SmartThings — платформа, що підтримує як локальне керування (через SmartThings Hub або платформу Edge), так і хмарну обробку даних. Система орієнтована на масовий ринок, має розвинений мобільний застосунок та інтерфейс автоматизації, сумісний з голосовими помічниками (Google Assistant, Bixby, Alexa). Платформа

підтримує широкий спектр протоколів: Zigbee[7], Z-Wave, Wi-Fi, Thread, Matter та Bluetooth. Відкритість до інтеграції сторонніх пристроїв обмежена рамками власницьких специфікацій Samsung, проте система залишається привабливою завдяки простоті встановлення та широкій підтримці брендових пристроїв (табл 1.3).

Таблиця 1.3 - Порівняльна характеристика платформ для розумного будинку

| Система | Архітект. | Протоколи підтримки | Тип конфігурації | Вартість | Локальне розг-ня | Відкритість |
|---|---|---|---|---|---|---|
| Home Assistant | Локальна | MQTT, Zigbee, Z-Wave, Wi-Fi, Thread, Matter | Веб-інтерфейс, YAML | Безкоштовна | Так | Відкрита (Apache 2.0) |
| Open HAB | Локальна (Java) | Zigbee, Z-Wave, EnOcean, Wi-Fi | Веб-інтерфейс, DSL | Безкоштовна | Так | Відкрита (EPL) |
| Domoticz | Локальна | Z-Wave, 1-Wire, MQTT, EnOcean, Wi-Fi | Веб-інтерфейс | Безкоштовна | Так | Відкрита (GPLv3) |
| Apple HomeKit | Гібридна | Bluetooth LE, Wi-Fi, Thread, Matter | Мобільний додаток iOS/macOS | Умовно безкоштовна | Так (з Apple-хабом) | Пропрієтарна |
| Samsung Smart Things | Гібридна | Zigbee, Z-Wave, Wi-Fi, Thread, Matter | Мобільний додаток, Edge | Безкоштовна | Так (з хабом) | Пропрієтарна |

Загалом, вибір платформи для побудови системи «розумного будинку» є багатофакторним процесом, що враховує технічні, економічні та експлуатаційні вимоги кінцевого користувача. Відкриті платформи (як-от Home Assistant,

openHAB, Domoticz) демонструють високий рівень гнучкості та адаптивності, водночас як комерційні рішення (Apple HomeKit, Samsung SmartThings) приваблюють інтеграцією в екосистеми, зручністю для кінцевого споживача та стабільністю.

Таким чином, оптимальний вибір інформаційної системи має базуватись на раціональному поєднанні функціональних можливостей, вартості впровадження, технічної сумісності з апаратними компонентами, а також відповідності вимогам інформаційної безпеки та конфіденційності даних.. За підсумками аналізу, найкращим рішенням для побудови інтелектуальної інформаційної системи «розумний дім» було обрано платформу Home Assistant, яка забезпечує оптимальний баланс між гнучкістю налаштування, широкою підтримкою пристроїв, активною спільнотою користувачів і розробників, а також високим рівнем контролю за збереженням даних у межах локальної мережі.

Вибір саме Home Assistant як основної платформи був зумовлений її здатністю задовольнити вимоги до локального виконання логіки автоматизації, що значно підвищує надійність роботи системи в умовах відсутності або нестабільності підключення до Інтернету. Крім того, платформа підтримує інтеграцію з широким переліком апаратних і програмних компонентів, зокрема сенсорами, виконавчими пристроями, мікроконтролерами (ESP8266/ESP32), протоколами зв’язку (MQTT, Zigbee, Z-Wave), а також дозволяє створювати складні сценарії керування за допомогою мови конфігурації YAML та шаблонізації Jinja2. Це надає розробникам високий рівень свободи у побудові гнучкої та адаптивної системи.

Окрім технічних переваг, важливою складовою вибору стала відкритість програмного забезпечення Home Assistant та активна спільнота, яка постійно підтримує розвиток проєкту, створює документацію та доповнення, ділиться рішеннями і практиками. Такий підхід сприяє швидшому впровадженню інновацій, ефективному усуненню помилок і розширенню функціональності системи. Усе це робить Home Assistant не лише технологічно придатною

платформою, а й перспективною основою для створення масштабованих та стабільних рішень у сфері автоматизації житлових приміщень.

## 1.5 Висновки

У процесі дослідження проаналізовано ключові технічні та організаційні аспекти побудови систем «розумного будинку», зокрема проблеми сумісності, безпеки, стандартизації та ефективності взаємодії між пристроями. Встановлено, що успішна реалізація подібних систем вимагає комплексного підходу до вибору програмної платформи, яка повинна забезпечувати масштабованість, гнучкість конфігурації, підтримку широкого спектра пристроїв, а також гарантувати контроль над обробкою і зберіганням даних.

На основі порівняльного аналізу сучасних рішень було оцінено переваги та недоліки існуючих інформаційних систем керування автоматизованими приміщеннями. Ураховуючи співвідношення функціональності, вартості, технічної сумісності та вимог до безпеки й конфіденційності, вибір було зосереджено на платформі Home Assistant. Ця система продемонструвала найкращу відповідність критеріям ефективної реалізації локальної клієнт-серверної архітектури для «розумного будинку», що й стало підґрунтям для її подальшого впровадження у межах даного проєкту.

# 2 ВИБІР ОПТИМАЛЬНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

## 2.1 Вибір оптимальної ІТ-інфраструктури

Для реалізації інформаційної системи «Розумний будинок» на базі локальної клієнт-серверної архітектури необхідно обґрунтовано обрати оптимальну ІТ-інфраструктуру, яка відповідатиме технічним вимогам, умовам експлуатації в межах однієї квартири та забезпечить необхідний рівень надійності, безпеки, масштабованості й енергоефективності. З урахуванням сучасних тенденцій та технічних можливостей, доцільним є впровадження системи на базі програмного забезпечення Home Assistant, яке характеризується відкритою архітектурою, активною спільнотою підтримки, широким спектром інтеграцій із пристроями різних виробників, а також здатністю до повністю локального функціонування без потреби у постійному підключенні до хмарних сервісів (рис. 2.1).

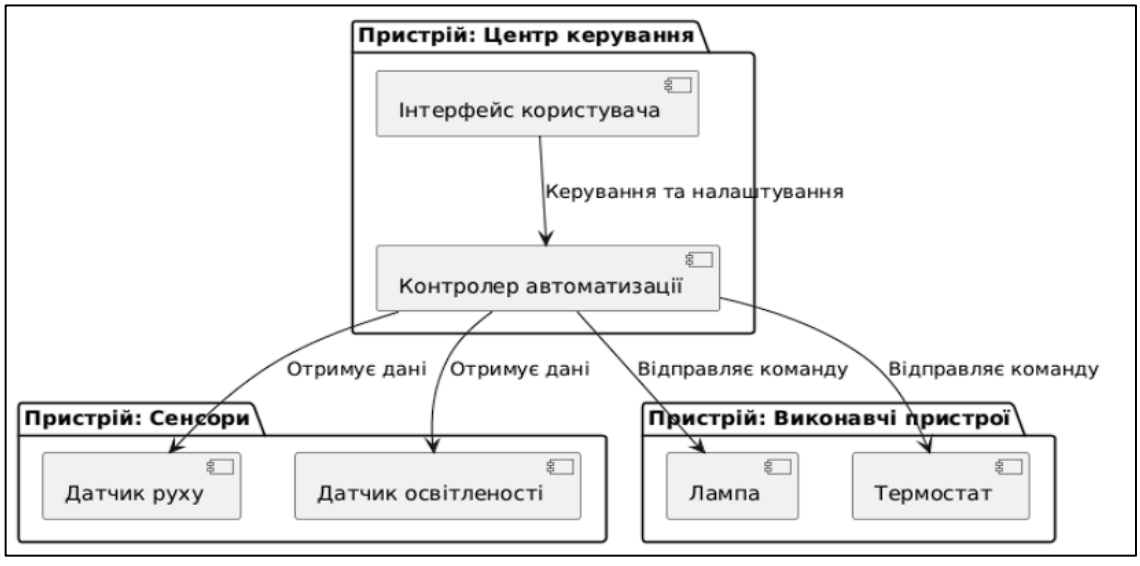


Рисунок 2.1 - Архітектурна модель системи «Розумний дім».

Як апаратну основу для серверної частини системи обґрунтовано обрано міні комп’ютер з обсягом оперативної пам’яті 4–8 ГБ, що є достатнім для обробки даних сенсорів, виконання автоматизацій та управління пристроями в

умовах однієї квартири. До його переваг належить низьке енергоспоживання (3–5 Вт у режимі очікування), компактність, наявність розширених можливостей підключення (USB, Ethernet, GPIO), а також повна підтримка Home Assistant. У разі потреби підвищення продуктивності (наприклад, для реалізації додаткових мультимедійних або відеоаналітичних функцій) доцільним є використання одноплатного комп'ютеру Raspberry Pi 4 (або 5). Такі системи забезпечують вищу обчислювальну потужність, швидкодію, надійне SSD-сховище та прийнятний рівень енергоспоживання (6–8 Вт у режимі простою), що забезпечує їхню придатність для розгортання високонавантажених локальних серверів.

Для забезпечення стабільного функціонування інформаційної системи рекомендовано використовувати гібридну мережеву інфраструктуру: провідне з'єднання (Ethernet) — для серверних компонентів та критичних вузлів, бездротове з'єднання (Wi-Fi) — для мобільних і автономних пристроїв. З метою підвищення безпеки та ізоляції інфраструктури «розумного будинку» від основної домашньої мережі доцільним є впровадження окремого SSID або VLAN для IoT-пристроїв. Усі процеси управління відбуваються локально, завдяки чому навіть у разі втрати підключення до Інтернету система зберігає працездатність.

Для обміну даними між пристроями та сервером використовуються спеціалізовані протоколи: Zigbee — енергоефективний mesh-протокол для сенсорів і виконавчих пристроїв; Z-Wave — аналогічна технологія з нижчою частотою (868 МГц), яка краще проходить через стіни та менш піддається перешкодам; Bluetooth — для близькодіапазонних пристроїв з низьким енергоспоживанням. Значну роль відіграє MQTT[3] — легкий протокол поверх TCP/IP, який забезпечує ефективну взаємодію між Wi-Fi-пристроями та шлюзами (наприклад, Zigbee2MQTT). Водночас безпосереднє використання Wi-Fi актуальне лише для пристроїв, які підключені до електромережі (камери, розетки, реле), оскільки цей тип з'єднання характеризується підвищеним енергоспоживанням.

У системі використовуються як сенсори (датчики температури, вологості, руху, освітленості, відкриття дверей/вікон) (рис. 2.2), так і виконавчі пристрої

(розетки, лампи, реле, замки, термостати). Поширеними виробниками є Aqara, Sonoff, IKEA, Shelly, Philips Hue. Для управління ІЧ/РЧ пристроями використовуються універсальні передавачі типу Broadlink RM або Sonoff RF Bridge. Завдяки відкритій архітектурі Home Assistant, користувач отримує змогу інтегрувати понад 2500 типів пристроїв та сервісів у єдине середовище управління, з можливістю створення кастомних сценаріїв автоматизації.

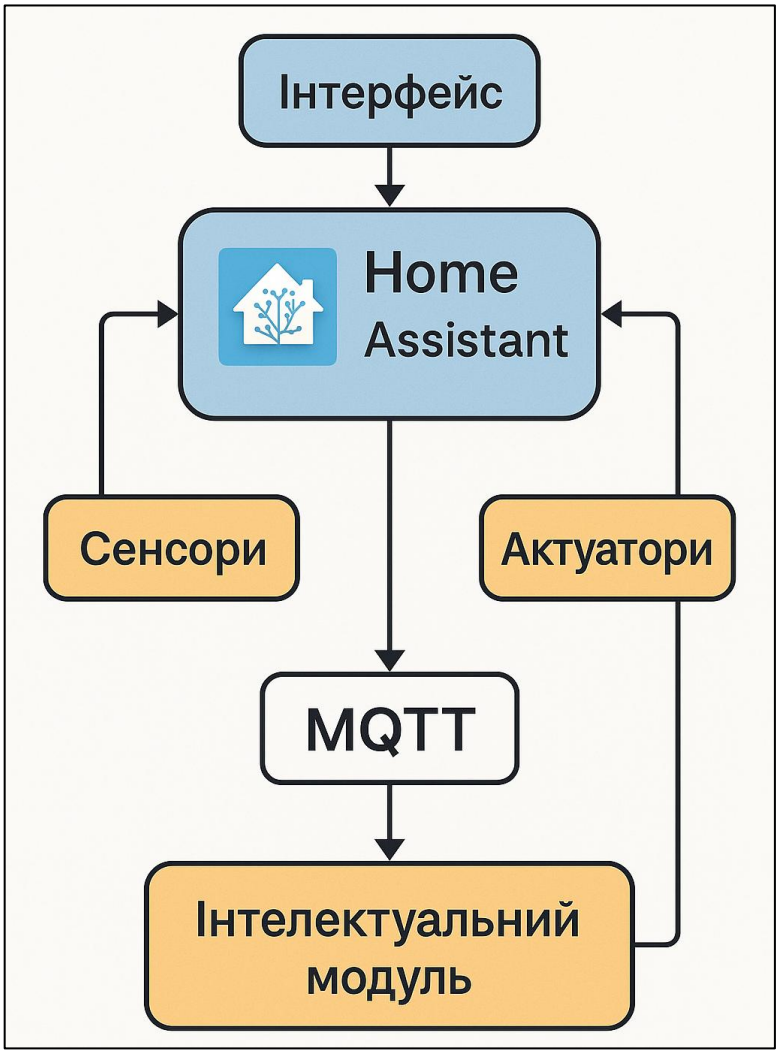


Рисунок 2.2 - Компоненти системи «розумний будинок» та принципи їхньої взаємодії

Однією з ключових вимог до системи є безпека. Задля цього впроваджуються сучасні методи захисту: шифрування даних (SSL/TLS), захищені мережеві протоколи (HTTPS, VPN), окремі IoT-сегменти мережі, надійна автентифікація, включно з двофакторною. Резервне копіювання

налаштувань системи виконується локально або з використанням хмарного сервісу Nabu Casa. Збереження історичних даних здійснюється на локальному сервері у вигляді бази даних (SQLite або PostgreSQL), що додатково забезпечує контроль над приватною інформацією користувача.Таким чином, вибрана IT-інфраструктура на основі локального сервера (Raspberry Pi або міні-ПК), гібридного мережевого середовища, спеціалізованих протоколів зв'язку (Zigbee, Z-Wave, MQTT, Wi-Fi), широкого спектра сенсорів і виконавчих пристроїв, а також із впровадженням сучасних методів безпеки, забезпечує ефективне, масштабоване та захищене функціонування системи «Розумний будинок» в умовах однієї квартири. Використання платформи Home Assistant є обґрунтованим та оптимальним вибором для реалізації завдань проєкту.

## 2.2 Аналіз можливостей мови програмування YAML

У сучасних інформаційних системах, зокрема в системах автоматизованого управління «розумним будинком», важливу роль відіграє вибір мови опису конфігурацій та структурованих даних. Одним із найбільш зручних і гнучких форматів для цієї мети є YAML[11] (Yet Another Markup Language, або YAML Ain't Markup Language). Це мова серіалізації даних, яка орієнтована на максимальну читабельність та легкість ручного редагування. Її синтаксис базується на відступах для відображення ієрархій, що дозволяє уникнути використання зайвих символів, таких як фігурні дужки чи теги, характерні для інших мов (наприклад, JSON або XML). YAML дозволяє описувати дані у вигляді простих пар «ключ–значення», списків та словників, підтримує вкладені структури, коментарі, посилання (anchors) та багато-документні файли. Завдяки цим можливостям YAML забезпечує як простоту для людини, так і достатню виразність для складних структур даних, що робить його придатним для опису конфігураційних файлів, сценаріїв автоматизації, параметрів пристроїв та логіки роботи в межах систем «розумного будинку».

У практичному застосуванні YAML набув широкого поширення в таких платформах, як Home Assistant, OpenHAB, Domoticz, де він використовується для

визначення конфігурацій MQTT-пристроїв, написання правил автоматизації, зберігання станів пристроїв тощо. Наприклад, за допомогою YAML можна описати сценарій ввімкнення освітлення о певній годині або за настанням події, такої як захід сонця. Файл конфігурації може містити блок automation, у якому за допомогою структурованих вкладень задаються умови спрацювання (триґери) та дії системи. Завдяки можливості використовувати коментарі (#) та лаконічному синтаксису, ці конфігурації легко читати, перевіряти і змінювати без необхідності глибоких технічних знань. Важливо також відзначити, що YAML підтримується широким спектром бібліотек і фреймворків у різних мовах програмування (Python, JavaScript, Java, Go тощо), що забезпечує зручність у побудові клієнт-серверних застосунків у межах локальної архітектури «розумного будинку».

При порівнянні YAML з іншими поширеними форматами серіалізації — JSON та XML — слід зазначити низку принципових відмінностей. YAML має найвищий рівень читабельності завдяки відсутності надлишкових синтаксичних елементів. На відміну від JSON, YAML підтримує коментарі, що критично важливо при створенні вручну редагованих конфігурацій. XML, хоч і забезпечує високу гнучкість завдяки використанню тегів і просторів імен, є надмірно об’ємним і менш придатним для сценаріїв, де ключовими є швидкість читання, зрозумілість та компактність. JSON, у свою чергу, зручний для машинної обробки, проте його синтаксис менш гнучкий і не підтримує такі особливості YAML, як анкори чи злиття мап. Табличне порівняння цих трьох форматів (YAML, JSON, XML) показує, що YAML є найбільш збалансованим з погляду читабельності, зручності підтримки, гнучкості опису та компактності параметрів, що особливо важливі при створенні конфігурацій у розподілених і локальних системах керування(табл. 2.1). Наприклад, для конфігурування пристрою, підключеного через MQTT, YAML дозволяє визначити такі параметри, як ім’я сенсора, адреса MQTT-топіку, властивості датчика тощо, у стислому та структурованому вигляді. Аналогічно, для визначення сценарію автоматизації, наприклад, увімкнення світла за годину до заходу сонця, YAML

дозволяє задати часовий триґер, цільовий пристрій, тип дії та додаткові параметри.

Таблиця 2.1 - Порівняння мов серіалізації даних для інформаційних систем керування

| **Критерій** | **YAML** | **JSON** | **XML** |
|---|---|---|---|
| Читабельність | Висока | Середня | Низька |
| Підтримка коментарів | Так | Ні | Частково (CDATA) |
| Компактність | Висока | Висока | Низька |
| Складність синтаксису | Низька | Середня | Висока |
| Придатність для ручного редагування | Висока | Середня | Низька |

Усе це — у зрозумілому, коментованому форматі, придатному як для ручного редагування, так і для автоматичної обробки серверною частиною системи. YAML-файли також зручно використовувати для зберігання параметрів, таких як температурні пороги, статуси пристроїв, ідентифікатори елементів тощо ( лістинг 2.1).

```
sensor:
  - platform: mqtt
    name: "Temperature Sensor"
    state_topic: "home/livingroom/temperature"
    unit_of_measurement: "°C"
    value_template: "{{ value_json.temperature }}"
```

Лістинг 2.1 – Приклад конфігурації MQTT-сенсора в YAML

Узагальнюючи, мову YAML доцільно розглядати як одну з оптимальних інформаційних технологій для створення систем «розумного будинку» на базі локальної клієнт-серверної архітектури. Вона поєднує зручність і гнучкість опису даних, адаптованість до ручного налаштування і масштабованість, що забезпечує ефективне розгортання, підтримку та модернізацію таких систем у реальному середовищі.

## 2.3 Аналіз програмної платформи Home Assistant

Home Assistant – це відкрита платформа для домашньої автоматизації, яку розроблено з акцентом на локальному керуванні і збереженні приватності користувача. Архітектура системи модульна: ядро реалізовано на Python 3, а функціонал розширюється численними інтеграціями та компонентами. Така модульна побудова дозволяє легко додавати підтримку нових пристроїв і сервісів – система побудована за модульним підходом, що спрощує реалізацію будь-яких розширень (рис. 2.3). Home Assistant вигідно відрізняється від багатьох конкурентів тим, що він може працювати повністю локально без залежності від хмарних сервісів, хоча при бажанні пропонується і безпечний віддалений доступ через VPN, SSL (Let’s Encrypt) або офіційний хмарний сервіс Home Assistant Cloud.

Конфігурація Home Assistant централізовано задається у файлах на основі YAML. Головний файл configuration.yaml містить список інтеграцій і їх параметри. Хоча більшість інтеграцій можна налаштовувати через веб-інтерфейс (через розділ Devices & Services), деякі з них досі потребують прямого редагування YAML-файлів.

У конфігурації також широко застосовуються можливості шаблонізації (Jinja2), включення зовнішніх файлів тощо, що підвищує гнучкість налаштування. У конфігурації також широко застосовуються можливості шаблонізації (Jinja2), включення зовнішніх файлів тощо, що підвищує гнучкість налаштування. Це дозволяє створювати динамічні сценарії автоматизації, адаптовані до змін середовища та поведінки користувача.

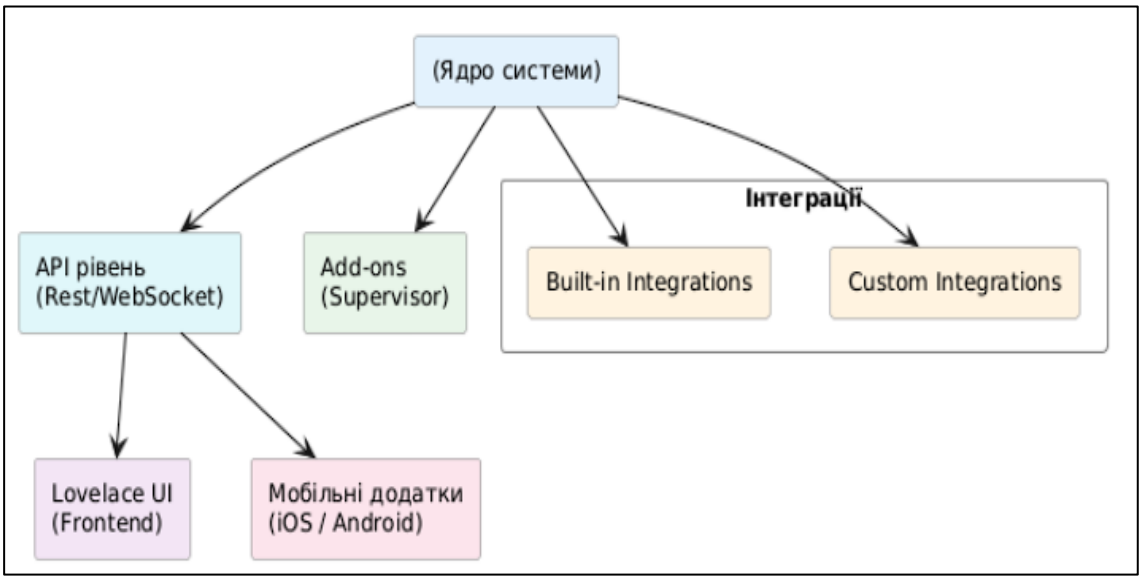


Рисунок 2.3 – Архітектура Home Assistant із розмежуванням ядра, інтеграцій, API та UI

Інтеграції в Home Assistant реалізовані на Python: кожна інтеграція відповідає за певну доменну область (наприклад, освітлення чи кліматконтроль), може опрацьовувати події, пропонувати дії і підтримувати внутрішній стан пристроїв. Home Assistant постачається з сотнями «built-in» інтеграцій на популярні пристрої та сервіси, а також підтримує створення власних кастомних компонентів за допомогою API розробника.

Крім того, для розширення екосистеми передбачені додатки (add-ons) – це окремі контейнери з додатковим програмним забезпеченням (наприклад, MQTT-брокери, вебсервери, системи відеоспостереження тощо), які можна встановлювати через вбудований Add-on Store Supervisor у версіях Home Assistant OS або Supervised. Доповнення дозволяють доповнити HA тими сервісами, які система самостійно не надає «з коробки», але можуть знадобитися у комплексній системі розумного дому. Зауважимо, що ці доповнення доступні лише в рамках офіційних методів інсталяції (HA OS/Supervised); в інших випадках аналогічні сервіси потрібно розгортати вручну.

Home Assistant пропонує можливості автоматизації. Автоматизації будуються на комбінації тригерів (подій), умов та дій, що дозволяє реагувати на зміни в системі без взаємодії користувача (рис. 2.4). Такий підхід забезпечує

гнучке та адаптивне керування пристроями, зменшуючи навантаження на користувача та підвищуючи загальний рівень комфорту.

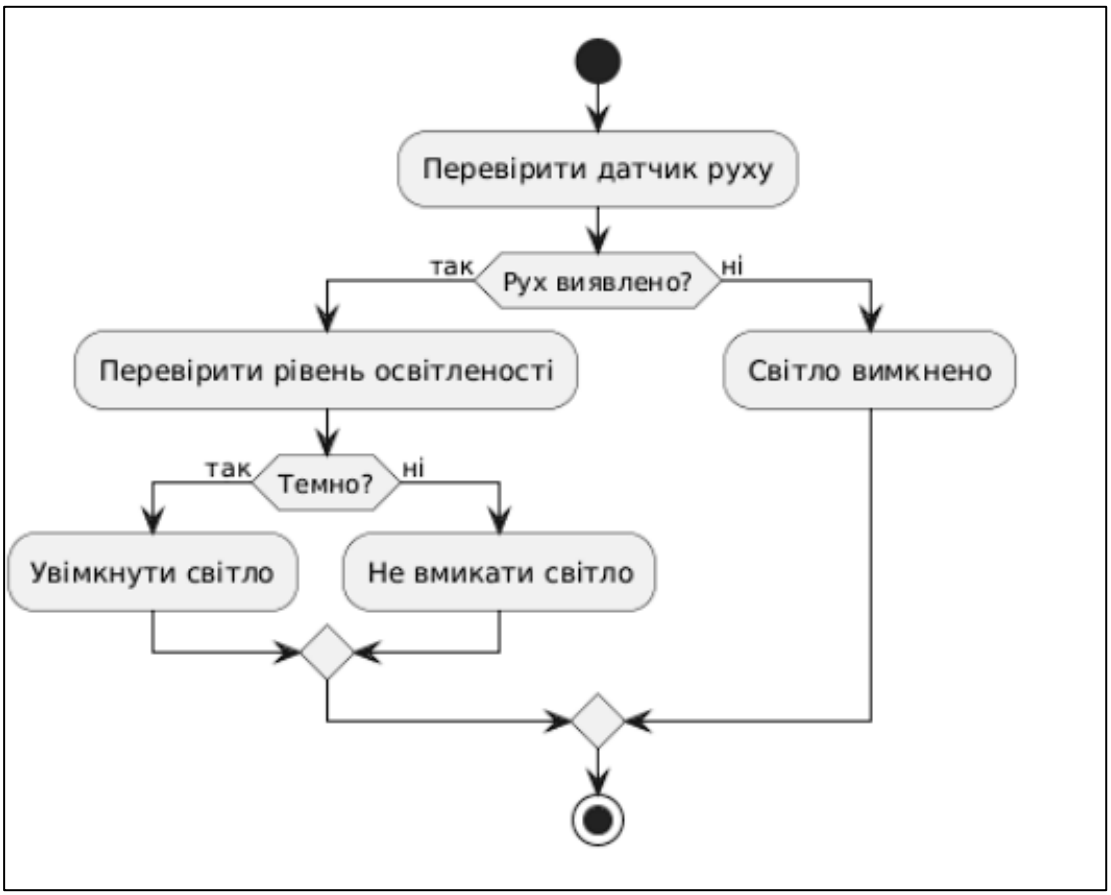


Рисунок 2.4 – Модель логіки автоматизації в Home Assistant

Наприклад, можна автоматично вмикати освітлення при заході сонця або вимикати музику під час телефонного дзвінка. Для спрощення створення типових сценаріїв впроваджено механізм Blueprints – готові шаблони автоматизацій, які можна імпортувати та налаштовувати без необхідності писати код з нуля. Крім стандартних автомацій, в HA можна створювати сцени – фіксовані набори станів кількох пристроїв (наприклад, «романтична сцена» фіксує конкретні положення світильників і кольори). Сцени зберігають бажані значення (on/off, яскравість, колір тощо) для групи сутностей, і їх можна активувати дією scene.turn_on. Крім того, система підтримує скрипти (scripts) – це послідовності дій, які виконуються за викликом. Сценарії (скрипти) у Home Assistant створюються у форматі YAML і дозволяють комбінувати будь-які

підтримувані дії (вмикання/вимикання пристроїв, затримки, логування тощо) у певному порядку. Скрипти можуть запускатися вручну, за кнопкою інтерфейсу або з іншої автоматизації, і розглядаються як окремий тип сутності, що виконує закладену послідовність.

Графічний інтерфейс Home Assistant (Lovelace UI) – це веб-керована консольна панель, що гнучко налаштовується користувачем. Користувач може створювати власні dashboard із різними видами карток та віджетів, задавати теми, іконки, групувати панелі по розділах і робити їх як статичними, так і адаптивними під мобільний вигляд. Усі ці можливості дають змогу зручно керувати домашньою автоматизацією як з десктопного браузера, так і з мобільного пристрою. Існують офіційні мобільні додатки Home Assistant Companion для iOS та Android, які надають доступ до тих самих панелей інтерфейсу, а також підтримують push-сповіщення і додаткові можливості мобільного пристрою (геолокація, акселерометр, камера тощо). Інтерфейс Lovelace цілком інтегрований з веб-технологіями, і для нього існують численні плагіни та користувацькі картки від спільноти.

Home Assistant постійно оновлюється — зазвичай команда випускає нові версії приблизно раз на місяць з новим функціоналом та виправленнями. Для інсталяцій з Supervisor оновлення ядра, системи та додатків здійснюються через веб-інтерфейс одним кліком. При цьому рекомендується регулярно підтримувати систему в актуальному стані з огляду на безпеку та стабільність. Для відновлення та захисту налаштувань НА передбачена вбудована система резервного копіювання (Snapshot/Backup). Спеціальна інтеграція «Backup» дозволяє створювати знімки всієї конфігурації та бази даних. Починаючи з версії 2025.1, Home Assistant реалізував можливість автоматичних резервних копій за розкладом: система сама регулярно зберігає бекапи, що спрощує дотримання практик безпеки. Крім того, для підписників Home Assistant Cloud додано опцію автоматичного зберігання зашифрованої копії у хмарі (до 5 ГБ) для надійного off-site зберігання. Щодо безпеки, то НА підтримує централізоване зберігання секретів у файлі secrets.yaml і закликає використовувати складні паролі та опціонально двофакторну аутентифікацію. Оскільки Home Assistant не потребує

хмари для основної роботи, дані зазвичай залишаються локальними, що підвищує приватність. Водночас для безпечного віддаленого доступу рекомендується SSL (Let’s Encrypt/DuckDNS) або Home Assistant Cloud. На офіційному сайті є сторінки Integration Alerts і Security Alerts, що інформують про знайдені уразливості в інтеграціях, а також сторінка System Status для загального стану платформи.

Порівняння з іншими популярними платформами показує відмінності у філософії та організації: openHAB[18] (Java-платформа) також гнучка і підтримує майже всі протоколи (Z-Wave, Zigbee, MQTT тощо), відома своєю розвиненою системою правил і кросплатформністю (працює на Windows, Mac, Linux). OpenHAB має велике співтовариство і орієнтована на технічно підкованих користувачів, але дещо складніша в налаштуванні. Domoticz – дуже легковажне C/C++ рішення з простим веб-інтерфейсом, орієнтоване на мінімальні апаратні вимоги і легкість використання. Domoticz споживає мало ресурсів і підходить для початківців, але порівняно з HA має обмеженіший набір інтеграцій і функцій. ioBroker – це платформа на базі Node.js, що робить ставку на велику кількість адаптерів (підтримка величезної кількості пристроїв та сервісів) і програмування правил на JavaScript. ioBroker дуже гнучкий у питанні інтеграцій, проте його інтерфейс менш інтуїтивний. У сумі Home Assistant вирізняється поєднанням «DIY»-підходу з дружнім графічним інтерфейсом і дуже активним співтовариством користувачів, що робить його одним із найпопулярніших рішень у своєму класі.

## 2.4 Архітектура системи: компоненти, принципи взаємодії, середовище виконання

Інформаційна система «розумний будинок» реалізована на основі локальної клієнт-серверної архітектури, яка забезпечує ефективну взаємодію між центральним керуючим пристроєм та периферійними модулями. У цій структурі серверну частину становить програмне забезпечення Home Assistant, розгорнуте на міні-комп’ютері MLLSE Mini PC M2 Air, на який попередньо

інстальовано Home Assistant OS — спеціалізовану операційну систему, розроблену для безперервної роботи в середовищі автоматизації житлових і комерційних приміщень. Клієнтську частину формують мікроконтролери ESP32, що функціонують під керуванням відкритої прошивки ESPHome, призначеної для спрощеного підключення та конфігурації сенсорних і виконавчих пристроїв.

Уся система умовно поділяється на кілька основних блоків (рис. 2.5):

- Фізичні сенсори — температурні датчики, датчики присутності, вологості, $CO_2$ тощо. Вони передають дані в реальному часі до центрального контролера.
- Актуатори / виконавчі пристрої — реле, розумні термоголовки або модулі керування котлом, які отримують команди від платформи й змінюють стан системи опалення.
- Центральна система управління — Home Assistant, розгорнутий на Raspberry Pi або локальному сервері, який приймає дані з сенсорів, обробляє їх і формує логіку дій.
- Інтелектуальний модуль — окремий скрипт або сервер (наприклад, Python з ML-моделлю), який може приймати дані від Home Assistant, аналізувати їх і повертати оптимальне рішення.
- Інтерфейс користувача — вебпанель Home Assistant, мобільний застосунок або голосове керування через Google Assistant чи інші сервіси.

Home Assistant OS побудована на основі ядра Linux і являє собою вбудований дистрибутив, орієнтований на енергоефективні комп’ютери з низьким енергоспоживанням.

Ця система включає повноцінне середовище для автономної роботи без участі хмарних сервісів: ядро Home Assistant, систему управління Supervisor, інтерфейс резервного копіювання, а також підтримку широкого спектра протоколів інтеграції. Завдяки цьому забезпечується високий рівень контролю над усіма процесами, що відбуваються у межах локальної мережі. Відсутність залежності від зовнішніх серверів знижує ризики витоку даних і підвищує стабільність роботи системи у разі відсутності підключення до Інтернету. Платформа дозволяє легко масштабувати рішення, додаючи нові пристрої та

функціональність без зміни базової архітектури. Крім того, активна спільнота розробників та регулярні оновлення сприяють постійному вдосконаленню функціоналу та підвищенню безпеки.

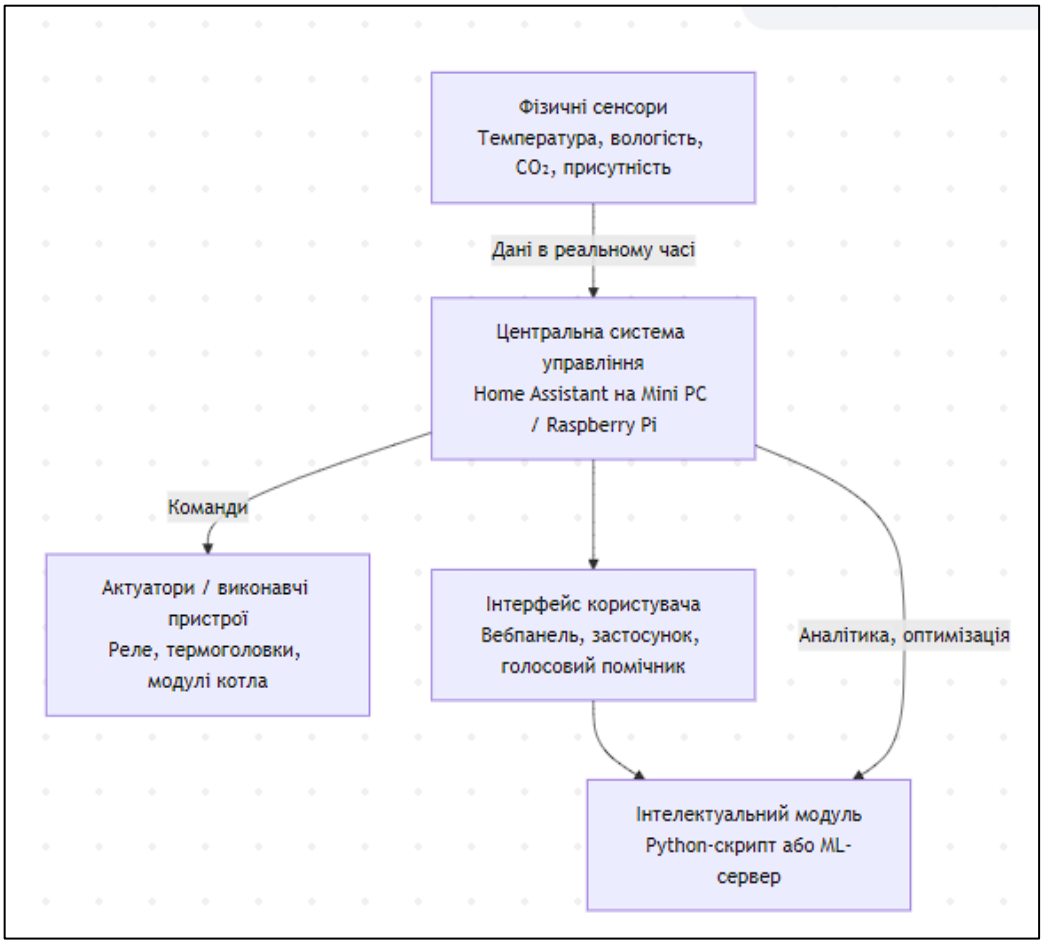


Рисунок 2.5 – Структурна схема інформаційної системи «Розумний будинок» на базі клієнт-серверної архітектури.

Ця система включає повноцінне середовище для автономної роботи без участі хмарних сервісів: ядро Home Assistant, систему управління Supervisor, інтерфейс резервного копіювання, а також підтримку широкого спектра протоколів інтеграції. Зокрема, вбудовано модулі для роботи з Zigbee, Z-Wave, MQTT, ESPHome Native API, Bluetooth Low Energy тощо. Таким чином, створюється централізоване ядро, здатне забезпечити безперервну координацію великої кількості компонентів розумного будинку у межах локального середовища.

Конфігурація системи здійснюється шляхом редагування конфігураційних файлів, переважно у форматі YAML. Це дозволяє забезпечити високу гнучкість налаштувань та зменшити залежність від графічного інтерфейсу. Розробники мають можливість конфігурувати як автоматизації, так і інтеграції з пристроями, прямо через текстові файли або ж за допомогою інтегрованого редактора File Editor. Крім того, у системі доступна підтримка таких інструментів, як Node-RED, що забезпечує побудову сценаріїв автоматизації у вигляді блок-схем, та AppDaemon, який дозволяє писати кастомні логіки автоматизації мовою Python.

Сервер Home Assistant виконує ключові функції інформаційної системи: обробку логіки автоматизації, збір і збереження даних від сенсорів, надання графічного інтерфейсу користувачу через веб-додаток та забезпечення каналу зв’язку з усіма периферійними пристроями. Такий підхід дозволяє зосередити обробку інформації в одному центрі, що спрощує керування системою і її діагностику.

Клієнтська частина базується на мікроконтролерах ESP32, які використовуються як універсальні вузли для підключення датчиків і реле. Завдяки платформі ESPHome, ці пристрої легко конфігуруються через YAML-файли, які описують необхідні параметри роботи: тип сенсора, поріг спрацювання, логіку зчитування або реагування. Після генерації прошивки вона завантажується на пристрій, після чого той автоматично виявляється сервером за допомогою ESPHome Native API з використанням Protocol Buffers — ефективного протоколу серіалізації даних. Це дає змогу уникнути складних ручних налаштувань, що робить систему доступною навіть для користувачів із базовими технічними знаннями.

Типовими прикладами таких клієнтських пристроїв є цифрові датчики температури та вологості DHT22, інфрачервоні датчики руху PIR, а також одноканальні або багатоканальні реле, які використовуються для вмикання освітлення або керування іншими електроприладами. Усі вони мають повну підтримку в середовищі ESPHome, що забезпечує мінімальний час на інтеграцію та налаштування. Це дозволяє швидко створювати стабільні та функціональні IoT-рішення з гнучким налаштуванням логіки взаємодії між пристроями.

Передача даних між сервером і мікроконтролерами здійснюється за допомогою локальної мережі Wi-Fi, побудованої на основі стандарту IEEE 802.11. Такий підхід дозволяє забезпечити достатню швидкість обміну інформацією, низьку затримку та стабільність з’єднання в умовах обмеженого простору (наприклад, у межах однієї квартири або будинку). Однією з важливих особливостей реалізованої архітектури є повна автономність — система не вимагає постійного з’єднання з Інтернетом для забезпечення базової функціональності. Для автоматичного виявлення нових пристроїв у локальній мережі використовується протокол mDNS (Multicast DNS), який дозволяє знаходити пристрої за ім’ям без необхідності фіксованих IP-адрес або складного налаштування маршрутизації.

Таким чином, побудована клієнт-серверна архітектура забезпечує надійний і масштабований фундамент для реалізації сучасної системи автоматизації житла, що об'єднує гнучке налаштування, простоту впровадження і високу надійність у щоденному використанні.

## 2.5 Висновки

У процесі реалізації системи було здійснено обґрунтований вибір IT-інфраструктури, здатної забезпечити стабільну та ефективну роботу системи «Розумний будинок» у межах локальної клієнт-серверної архітектури. Обрана конфігурація враховує вимоги до продуктивності, енергоефективності, масштабованості та захисту даних, з урахуванням специфіки взаємодії з пристроями реального часу.

Особливу увагу приділено аналізу мови опису конфігурацій — YAML, яка відіграє ключову роль у налаштуванні логіки автоматизації в системі Home Assistant. Простий синтаксис, читабельність і широкі можливості для структурування складних сценаріїв автоматизації роблять YAML придатним інструментом як для початкової розробки, так і для подальшої підтримки системи. Завдяки гнучкості YAML, розробники можуть ефективно керувати конфігурацією, легко вносити зміни та адаптувати систему до нових вимог.

Детально проаналізовано функціональні можливості платформи Home Assistant, що стала базовою програмною основою системи. Серед ключових переваг: підтримка великої кількості інтеграцій, активна спільнота розробників, регулярні оновлення, модульна структура та підтримка локального виконання сценаріїв без залежності від сторонніх хмарних сервісів.

Було розроблено архітектуру системи, що охоплює програмні та апаратні компоненти, середовище виконання та принципи взаємодії між модулями. Забезпечено логічний поділ на рівні збору даних, обробки, зберігання та управління пристроями. Така структура дозволяє гнучко адаптувати систему до змін у конфігурації, додавати нові модулі, а також підтримувати високий рівень автономності та стабільності в роботі.

# 3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОБОТИ КОРИСТУВАЧІВ МЕРЕЖІ «РОЗУМНИЙ БУДИНОК»

## 3.1 Розроблення архітектури інформаційної системи

Система «розумний будинок» – це спеціалізована інформаційна система, яка інтегрує датчики (сенсори) та виконавчі пристрої (актуатори) у межах локальної мережі житлової будівлі, забезпечуючи їх дистанційний моніторинг і управління. Архітектура системи реалізована за класичною клієнт-серверною моделлю: дані від датчиків передаються через локальну мережу на центральний сервер (або хаб), який обробляє цю інформацію і зберігає її в базі даних. На сервері реалізовано бізнес-логіку системи та алгоритми управління – саме він генерує команди, які надходять до виконавчих пристроїв.

Клієнтськими інтерфейсами системи виступають мобільні додатки або веб-інтерфейси на комп’ютері, що формують запити до сервера. У класичній клієнт-серверній архітектурі клієнти запитують дані або послуги у сервера, а сервер надає необхідні ресурси (зокрема, дані з бази чи контрольні команди). Це дозволяє централізовано управляти всіма підсистемами будинку: сервер відповідає за узгодження роботи сенсорів і актуаторів, оновлення станів пристроїв і своєчасне реагування на події. Такий розподіл ролей підвищує продуктивність і безпеку системи, оскільки дані централізовано зберігаються та обробляються в одному місці. Ключовим апаратним елементом є контролер (хаб), або «мозок» системи: він об’єднує всі пристрої в єдину мережу і забезпечує зв’язок між сенсорами, сервером і користувачем. Датчики (температури, вологості, світла, руху, диму, затоплення тощо) фіксують параметри навколишнього середовища і передають виміряні значення на центральний контролер чи сервер. Актуатори (виконавчі пристрої: розумні лампи, реле, електрозамки, клапани, сирени тощо) отримують команди від серверу/контролера і змінюють фізичний стан системи (нагрів теплоносія, вмикання світла, блокування дверей, подання сигналу тривоги тощо). Усі вхідні дані від датчиків записуються в базу даних системи для подальшого аналізу та

ведення журналу подій, що забезпечує можливість аудиту і візуалізації історії роботи «Розумного будинку»

Наприклад, підсистема освітлення включає датчики руху або освітленості та керовані освітлювальні прилади (димери, світлодіоди, розумні лампи), що дозволяє автоматично регулювати світло за заданими сценаріями або залежно від руху людей. Підсистема безпеки містить датчики проникнення (відчинення вікон/дверей), детектори диму/газу, відеокамери та сигнали тривоги (сирени), які активуються при аномальних подіях. Ці підсистеми інтегровані в єдину клієнт-серверну платформу: сервер обробляє їхні сигнали згідно з налаштованими правилами й сценаріями, а у разі загрози запускає механізми оповіщення. База даних гарантує надійне збереження налаштувань і станів пристроїв для забезпечення безперервної роботи та безпеки системи (рис. 3.1).


Рисунок 3.1 – Структурна схема архітектури інформаційної системи «розумний будинок»

## 3.2 Розроблення алгоритму роботи платформи Home Assistant

Алгоритм функціонування платформи Home Assistant ґрунтується на подієво-орієнтованій архітектурі, в якій центральним елементом виступає шина подій (event bus). Усі компоненти системи взаємодіють між собою через публікацію та обробку подій. Основними складовими ядра платформи є: шина подій, машина станів (state machine), таймер, а також реєстр сервісів. Зміна стану будь-якої сутності (наприклад, спрацьовування датчика руху або вмикання

світильника) генерує подію state_changed, яка передається усім зацікавленим компонентам системи. Машина станів зберігає поточні значення всіх сутностей, оновлюючи їх відповідно до нових подій.

Користувач може реалізовувати складні логічні зв’язки за допомогою механізму автоматизацій. Автоматизація у Home Assistant складається з трьох ключових елементів: тригеру (trigger), умов (condition) та дій (action). Тригер визначає подію або стан, який ініціює виконання сценарію. Умова є необов’язковим елементом, який дозволяє додатково перевірити контекст (наприклад, час доби або наявність людей у будинку). Дія описує безпосередньо ті операції, які має виконати система – від вмикання пристроїв до надсилання повідомлень.

Для ілюстрації розглянемо типовий сценарій автоматизації, що передбачає керування освітленням за допомогою датчика руху. Логіка полягає у ввімкненні світильника в коридорі в разі фіксації руху після заходу сонця. Опис YAML-конфігурації виглядає наступним чином (лістинг 3.1):

```
automation:
  - alias: "Автоматичне ввімкнення світла в коридорі"
    trigger:
      - platform: state
        entity_id: binary_sensor.motion_hallway
        to: 'on'
    condition:
      - condition: sun
        after: sunset
    action:
      - service: light.turn_on
        target:
          entity_id: light.hallway
```

Лістинг 3.1- Фрагмент коду автоматизації ввімкнення освітлення в коридорі.

У цьому прикладі автоматизація активується при зміні стану датчика руху на «увімкнено». Умова перевіряє, чи поточний час пізніше заходу сонця. Якщо

обидві умови виконуються, система подає команду на увімкнення світильника в коридорі.

Інший приклад – автоматизація кліматичного контролю. Система може відслідковувати температуру в кімнаті та вмикати охолодження, якщо температура перевищує заданий поріг (лістинг 3.2):

```
automation:
  - alias: "Увімкнення кондиціонера при високій температурі"
    trigger:
      - platform: numeric_state
        entity_id: sensor.temperature_livingroom
        above: 25
    action:
      - service: climate.set_hvac_mode
        target:
          entity_id: climate.livingroom_ac
        data:
          hvac_mode: cool
```

Лістинг 3.2 - Конфігурація сценарію керування температурою.

У цьому випадку система реагує на перевищення температури вище 25 °C, після чого активує кондиціонер у режимі охолодження.

Автоматизація системи безпеки є ще одним прикладом застосування алгоритмів на базі Home Assistant. Наприклад, якщо система перебуває в режимі охорони і двері будинку відкриваються, можна реалізувати сповіщення користувача та активацію світлової сигналізації (лістинг 3.3):

Цей сценарій передбачає перевірку режиму сигналізації та відповідну реакцію при фіксації вторгнення. Усі автоматизації в Home Assistant реалізуються за єдиним принципом «тригер – умова – дія». Важливо, що кожен компонент системи діє у межах загальної подієвої моделі, що забезпечує гнучкість та масштабованість платформи. YAML-формат конфігурацій дозволяє

точно визначати поведінку системи без необхідності глибоких знань у програмуванні.

```
automation:
  - alias: "Попередження про вторгнення"
    trigger:
      - platform: state
        entity_id: binary_sensor.door_front
        to: 'on'
    condition:
      - condition: state
        entity_id: alarm_control_panel.home_alarm
        state: 'armed_away'
    action:
      - service: notify.mobile_app
        data:
          message: "Увага! Виявлено відкриття дверей у режимі охорони!"
      - service: light.turn_on
        target:
          entity_id: light.entryway
```

Лістинг 3.3 - Приклад автоматизації безпеки на платформі Home Assistant.

Отже, алгоритм роботи Home Assistant базується на безперервному моніторингу подій, аналізі умов та реалізації відповідних сценаріїв керування. Такий підхід забезпечує високий рівень автоматизації побутових процесів і дозволяє користувачу створювати гнучкі та адаптивні сценарії відповідно до своїх потреб.

## 3.3 Вибір апаратного та програмного забезпечення

Для реалізації системи «розумний будинок» обрано рішення, яке поєднує простоту розгортання, гнучкість конфігурації та відповідність вимогам локального управління без залежності від хмарних сервісів. У якості апаратної основи було використано компактний міні-ПК MLLSE Mini PC M2 Air (рис. 3.2),

який виконує функції центрального вузла системи. Завдяки 64-розрядному процесору, підтримці сучасних протоколів комунікації та наявності SSD-накопичувача, даний пристрій забезпечує необхідний рівень стабільності, продуктивності та безперервної роботи. На ньому інстальовано Home Assistant OS — спеціалізовану операційну систему, розроблену для побудови систем автоматизації з інтегрованим веб-інтерфейсом та підтримкою численних додатків і служб.

Рисунок 3.2 - Вигляд міні-ПК MLLSE M2

У запропонованій архітектурі основним інструментом взаємодії з фізичними пристроями є платформа ESPHome, що забезпечує генерацію

прошивок для мікроконтролерів ESP32/ESP8266 (рис. 3.3) на основі конфігураційних YAML-файлів

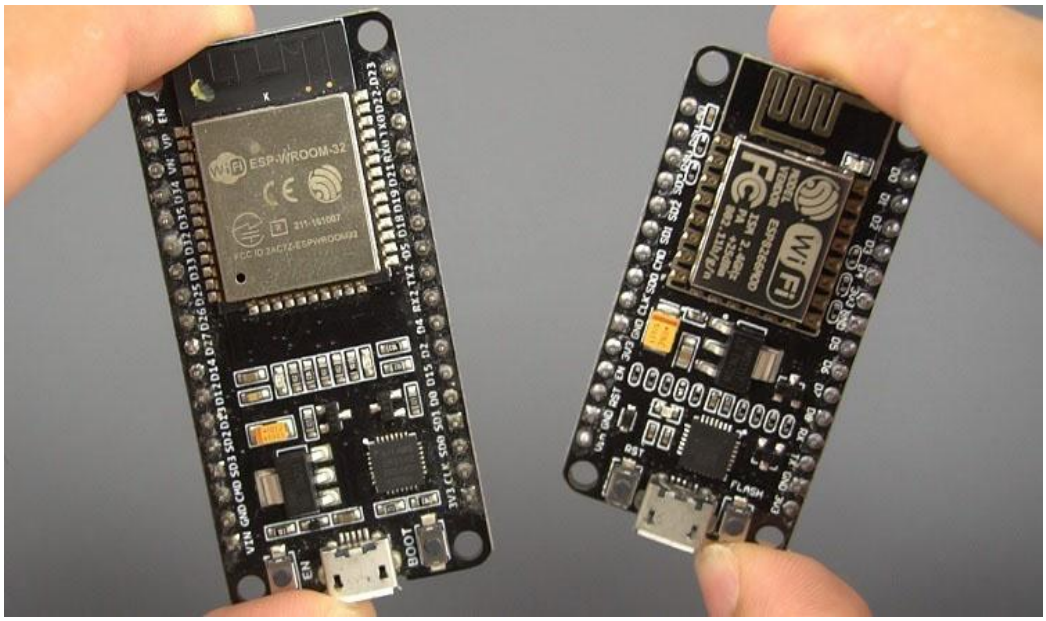


Рисунок 3.3 - Вигляд контролерів ESPHome

Це дозволяє швидко підключати різноманітні сенсори та виконавчі механізми без потреби глибокого програмування. Однією з головних переваг ESPHome є її повна інтеграція з Home Assistant, що забезпечує безшовну передачу даних між пристроями і головним сервером через локальну мережу.

Для підвищення функціональної гнучкості та масштабованості системи доцільним є використання додаткових програмних компонентів. Так, MQTT-брокер (наприклад, Mosquitto) може бути інтегрований як основний механізм обміну повідомленнями між пристроями, що використовують MQTT-протокол. Це особливо важливо у випадку майбутнього розширення системи сторонніми модулями. Редактор конфігурацій File Editor спрощує керування текстовими файлами конфігурацій без необхідності підключення через сторонні середовища розробки. Крім того, Node-RED, який пропонує графічне середовище для побудови логіки автоматизації, може бути залучений для створення складніших сценаріїв взаємодії. У разі інтеграції пристроїв стандарту Zigbee доцільно використовувати шлюз Zigbee2MQTT, який забезпечує взаємодію з Zigbee-

пристроями через MQTT-брокер та дозволяє уникати використання закритих екосистем (рис. 3.4).

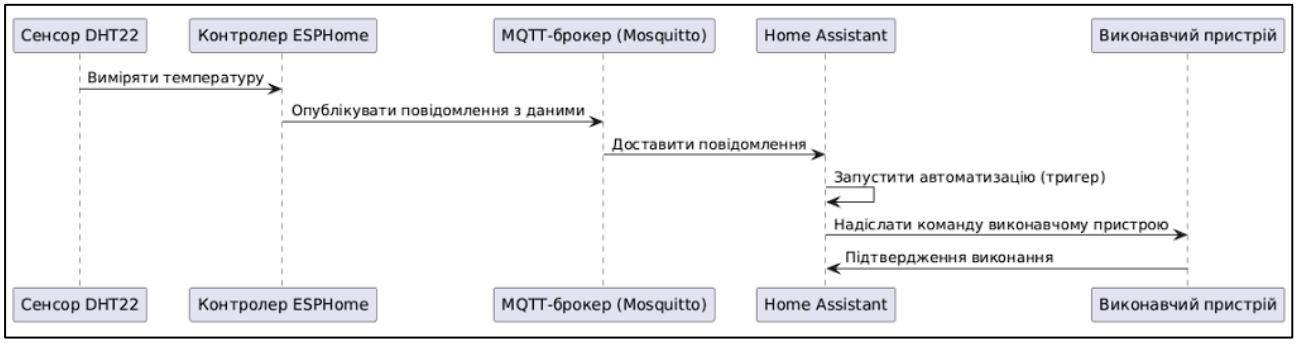


Рисунок 3.4 - UML-діаграма послідовності обміну даними між сенсором, контролером ESPHome, MQTT-брокером та платформою Home Assistant

Щодо апаратних пристроїв, до системи інтегруються найбільш поширені та перевірені компоненти. Наприклад, цифровий сенсор температури та вологості DHT22 дозволяє здійснювати моніторинг мікроклімату в приміщенні. Для виявлення присутності людей застосовуються пасивні інфрачервоні датчики руху (PIR), які генерують цифровий сигнал при фіксації руху в зоні покриття. У ролі виконавчих пристроїв використовуються керовані реле, здатні комутувати освітлення або інші електричні навантаження за сигналом із контролера. Усі згадані компоненти сумісні з ESPHome та можуть бути легко інтегровані в загальну інфраструктуру.

Обґрунтування вибору саме таких апаратних і програмних компонентів зумовлено їх відкритістю, активною підтримкою спільноти, широкою документаційною базою та надійністю при експлуатації у локальному середовищі. Комплексне використання Home Assistant, ESPHome та обраного міні-ПК створює стабільну, масштабовану та захищену платформу для реалізації інтелектуальної системи автоматизації житлового приміщення. Це дозволяє не лише забезпечити базову функціональність, а й закладає фундамент для подальшого розвитку системи з мінімальними витратами на її модернізацію. Такий підхід сприяє гнучкому масштабуванню та швидкому впровадженню

нових компонентів і технологій без необхідності суттєвих змін у вже наявній архітектурі.

## 3.4 Програмна реалізація системи розумного будинку

Програмна реалізація системи розумного будинку побудована на основі локальної клієнт-серверної архітектури з використанням міні-комп’ютера MLLSE M2 Air як серверної частини та мікроконтролерів ESP32, що функціонують як клієнтські пристрої. Сервер виконує роль централізованого контролера, на якому встановлено систему Home Assistant — програмну платформу з відкритим вихідним кодом, призначену для локального керування пристроями автоматизації. Клієнтські пристрої працюють під керуванням прошивки ESPHome, що забезпечує інтеграцію сенсорів, реле та інших компонентів у єдину систему.

Комунікація між сервером і клієнтськими пристроями здійснюється через локальну мережу за допомогою вбудованого ESPHome API або, за необхідності, протоколу MQTT. Такий підхід дозволяє повністю уникнути залежності від хмарних сервісів, забезпечуючи автономність, підвищену стабільність та конфіденційність обробки даних.

Налаштування всієї системи здійснюється за допомогою конфігураційних файлів у форматі YAML. У Home Assistant файли використовуються для визначення інтеграцій, автоматизацій та інтерфейсу користувача, тоді як в ESPHome – для опису апаратних підключень, сенсорів, виходів та логіки роботи пристроїв. Система підтримує широке коло функцій, серед яких керування освітленням, контроль температурного режиму, охоронні функції, управління доступом та сповіщення користувача.

Сервер на базі Home Assistant виконує централізовану обробку подій та сценаріїв. Після підключення ESPHome-пристрою Home Assistant автоматично виявляє та реєструє всі його сутності (сенсори, реле, виконавчі механізми). Усі подальші дії, зокрема автоматизації, можуть бути реалізовані як через графічний інтерфейс, так і шляхом прямого редагування YAML-конфігурацій. Це надає

гнучкість у налаштуванні логіки поведінки системи відповідно до потреб користувача.

Клієнтські пристрої — ESP32 із прошивкою ESPHome — виконують безпосередню роботу з датчиками та виконавчими пристроями. Наприклад, температурні датчики (DHT22, DS18B20) або датчики вологості описуються через платформу sensor (лістинг 3.4), а реле для керування освітленням — через компоненти switch або light (лістинг 3.5).

```
sensor:
  - platform: dht
    pin: GPIO4
    temperature:
      name: "Room Temperature"
    humidity:
      name: "Room Humidity"
    model: DHT22
    update_interval: 60s
```

Лістинг 3.4 – YAML-конфігурація сенсора температури та вологості DHT22 в ESPHome

```
switch:
  - platform: gpio
    pin: GPIO12
    name: "Relay Light"
```

Лістинг 3.5 – Приклад конфігурації реле для керування освітленням в ESPHome

Після компіляції YAML-файлу прошивка завантажується на ESP32, після чого пристрій функціонує автономно, підтримуючи з’єднання з сервером і забезпечуючи обмін даними в реальному часі(лістинг 3.6).

До основних функцій системи належить управління освітленням, реалізоване через керування реле або світлодіодними контролерами, що підключені до ESP32. Температурне регулювання здійснюється через інтеграцію температурних сенсорів і реле або керування кліматичними пристроями.

```
esphome:
  name: climate_node
  platform: ESP32
  board: esp-wrover-kit

wifi:
  ssid: "SmartHomeWiFi"
  password: "password123"
```

Лістинг 3.6 – Основна конфігурація ESP32-пристрою для роботи в мережі Wi-Fi

Безпекові функції реалізовано за допомогою датчиків руху, відкриття дверей і вікон, а також сирен. Усі ці пристрої можуть бути пов’язані з автоматизаціями, що виконуються Home Assistant на підставі отриманих тригерів (лістинг 3.7).

```
automation:
  - alias: "Turn on light when motion detected"
    trigger:
      platform: state
      entity_id: binary_sensor.motion_sensor
      to: "on"
    action:
      service: switch.turn_on
      target:
        entity_id: switch.relay_light
```

Лістинг 3.7 – Приклад автоматизації увімкнення світла при спрацюванні датчика руху в Home Assistant

Контроль доступу забезпечується керуванням електромагнітними замками, які підключаються до ESP32 через відповідні реле. Можливе розширення функціоналу через використання RFID-зчитувачів або кодових клавіатур, які дозволяють здійснювати аутентифікацію користувача локально. Автоматизація дій, пов’язаних із доступом, виконується за допомогою логічних умов і сценаріїв у конфігураціях Home Assistant.

Система підтримує надсилання повідомлень про події користувачу. Це можливо за рахунок вбудованих засобів Home Assistant, зокрема мобільного застосунку, через який користувач отримує push-сповіщення про виявлений рух, зміну температури або відкриття дверей. Конфігурації для надсилання повідомлень також описуються у YAML-форматі та інтегруються з іншими сценаріями.

Таким чином, реалізація програмної частини системи розумного будинку забезпечує повну інтеграцію апаратних компонентів з програмною логікою, централізоване управління через Home Assistant, а також модульну побудову, яка легко масштабується та адаптується до різних сценаріїв застосування. Використання відкритих протоколів, автономного функціонування та зрозумілої конфігурації на основі YAML дозволяє забезпечити як гнучкість, так і надійність роботи системи.

## 3.5 Роль аналізу даних у реалізації інтелектуальних сценаріїв керування системою «розумний будинок»

Один із ключових етапів реалізації інтелектуальної інформаційної системи «Розумний будинок» полягає в інтеграції модуля аналізу даних, що забезпечує автоматизоване та адаптивне керування різноманітними процесами на основі показників, зібраних від сенсорних пристроїв. У межах побудови такої системи джерелами даних виступають сенсори температури, вологості, освітленості, руху, рівня $CO_2$ та інші фізичні вимірювачі, що встановлюються у контрольованому середовищі. Отримані показники надсилаються через локальну мережу на центральний сервер, де відбувається обробка, аналіз та прийняття відповідних керуючих рішень.

Для забезпечення базової інтелектуальності системи обрано простий rule-based підхід, реалізований за допомогою автоматизацій на мові YAML у середовищі Home Assistant. Такий метод дозволяє створювати логічні умови типу «якщо–то», завдяки чому навіть без залучення складних алгоритмів машинного навчання система здатна автономно реагувати на зміни в

навколишньому середовищі. Наприклад, у разі перевищення заданого порогу температури може вмикатися система кондиціонування, або при спрацюванні датчика руху — автоматично активується освітлення в приміщенні. Подібна логіка дозволяє формувати адаптивну поведінку системи відповідно до сценаріїв, заздалегідь визначених користувачем.

Інтеграція модуля аналізу даних здійснюється у межах локальної клієнт-серверної архітектури, де серверна частина відповідає за обробку показників сенсорів та формування команд для виконавчих пристроїв. Основний процес побудований за принципом «сенсор–сервер–актуатор». У цій моделі сенсорний вузол фіксує зміни в навколишньому середовищі, серверна частина проводить аналіз отриманих даних згідно із заданими умовами, після чого передається відповідна команда до виконавчого модуля (наприклад, реле, лампа, вентилятор тощо).

Конфігурація подібної системи може бути реалізована за допомогою простих YAML-файлів у середовищі ESPHome, яке інтегрується з Home Assistant. Зокрема, налаштування включає опис сенсорів, бінарних датчиків та виконавчих елементів, а також автоматизації, що зв’язують умови з діями.sensor (лістинг 3.8):

Наприклад, у разі, якщо бінарний датчик руху змінює стан на «on», активується команда на вмикання світла у відповідному приміщенні. Цей підхід демонструє, як за допомогою доступних технологій та мінімальних обчислювальних ресурсів можливо реалізувати базовий рівень інтелектуального керування в системі розумного будинку.

Застосування подібного модуля аналізу дозволяє не лише автоматизувати побутові процеси, а й забезпечити підвищення енергоефективності, зручності користування та адаптивності системи до потреб мешканців. За допомогою інтеграції з сенсорами та виконавчими пристроями система може ефективно реагувати на зміни в оточуючому середовищі, оптимізуючи витрати енергії. Модуль аналізу даних дозволяє не тільки визначати поточний стан приміщення, а й передбачати майбутні потреби, надаючи можливість для прогнозованої адаптації. Це забезпечує не лише економію ресурсів, а й покращує комфорт

користувачів, надаючи їм більш інтуїтивно зрозуміле та персоналізоване керування. В результаті, система стає більш розумною, автономною та ефективною, що забезпечує зручність користування та підвищує загальний рівень якості життя мешканців.

```
  - platform: dht
    pin: D5
    temperature:
      name: "Температура в кімнаті"
    humidity:
      name: "Вологість в кімнаті"
binary_sensor:
  - platform: gpio
    pin: D7
    name: "Датчик руху"
    device_class: motion
light:
  - platform: gpio
    pin: D1
    name: "Лампа в кімнаті"
automation:
  - alias: "Увімкнути світло при виявленні руху"
    trigger:
      platform: state
      entity_id: binary_sensor.д_руху
      to: 'on'
    action:
      service: light.turn_on
      entity_id: light.лампа_в_кімнаті
```

Лістинг 3.8 – YAML-конфігурація автоматизації на основі датчика руху в Home Assistant.

Простота впровадження, відкритість платформи Home Assistant та підтримка великої кількості сенсорів і виконавчих пристроїв роблять даний підхід доцільним для реалізації в рамках локальної архітектури розумного житла (рис. 3.6).

Завдяки можливості локального зберігання та обробки даних, система на базі Home Assistant забезпечує високий рівень приватності та контролю над

інформацією, що особливо важливо в умовах зростання кіберзагроз. Вбудована підтримка різноманітних стандартів зв'язку, таких як MQTT, Zigbee, Bluetooth та Wi-Fi, дозволяє інтегрувати пристрої від різних виробників у єдину екосистему. Це значно розширює функціональні можливості системи, сприяє підвищенню надійності та дозволяє реалізовувати складні сценарії керування без залучення сторонніх хмарних сервісів. Такий підхід відповідає сучасним вимогам до гнучких, захищених та ефективних систем автоматизації побуту.

Крім того, використання локальної обчислювальної інфраструктури знижує затримки у виконанні команд, що є критичним для задач реального часу, таких як безпека або аварійне реагування. Зменшення залежності від зовнішніх хмарних сервісів дозволяє забезпечити миттєву реакцію на зміни в середовищі, що є важливим для захисту житлових приміщень та оперативного реагування на потенційні загрози. Це дозволяє значно підвищити надійність системи, оскільки навіть у разі відсутності підключення до Інтернету вона продовжує функціонувати без перебоїв. Така автономність робить систему більш стійкою до зовнішніх впливів та несприятливих умов.

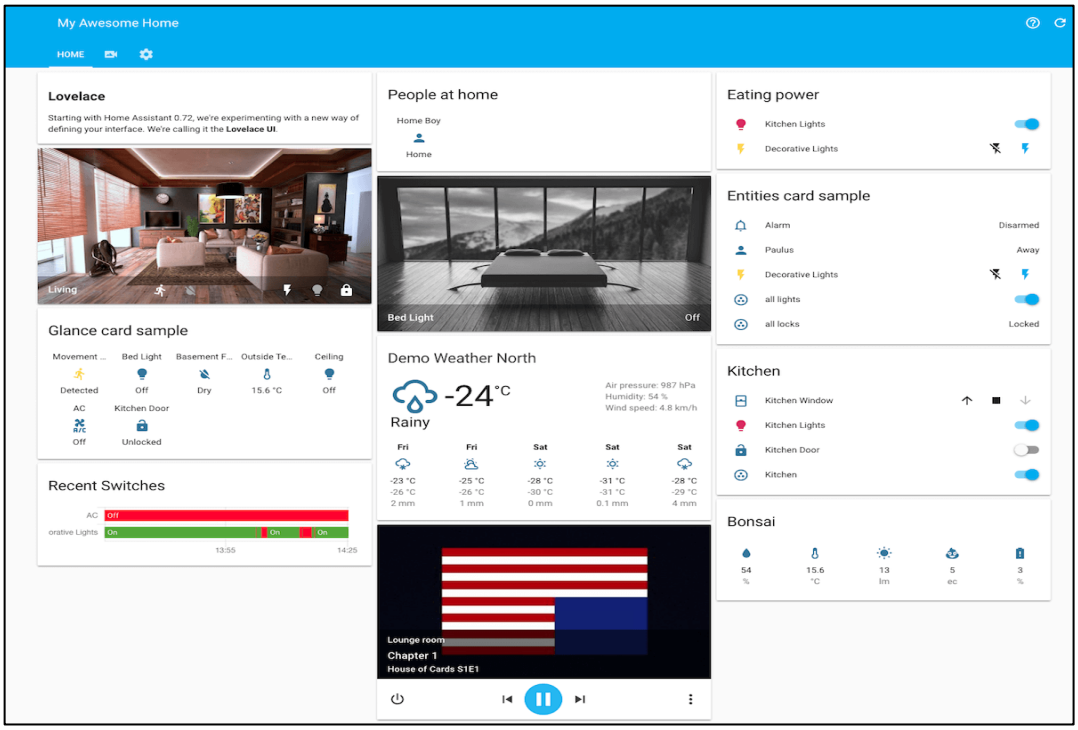


Рисунок 3.6 - Приклад користувацького інтерфейсу для моніторингу сенсорів у Home Assistant

## 3.6 Висновки

У результаті розробки інформаційної системи для керування елементами «розумного будинку» реалізовано комплексне рішення, що поєднує ефективну архітектуру, зручну логіку автоматизації та адаптивне керування на основі аналізу даних.

Запропонована архітектура побудована на основі локальної клієнт-серверної моделі, яка забезпечує високу швидкодію, стабільність і безпеку взаємодії між пристроями. Алгоритм роботи системи враховує як дані від сенсорів, так і дії користувача, забезпечуючи динамічне реагування на зміни в середовищі.

Програмну реалізацію здійснено на основі платформи Home Assistant, яка дозволяє легко інтегрувати апаратні компоненти та гнучко налаштовувати сценарії керування через конфігурацію YAML. Це дало змогу реалізувати інтуїтивно зрозуміле керування освітленням, температурою, вентиляцією та іншими побутовими функціями. У процесі розробки здійснено обґрунтований вибір апаратного забезпечення, зокрема мікроконтролерів, сенсорів та актуаторів, які відповідають вимогам щодо продуктивності, енергоефективності та сумісності з обраною платформою.

Інтеграція модуля аналізу даних на основі показників сенсорів забезпечила інтелектуальну поведінку системи — автоматичне керування пристроями за умовами, наближеними до реальних сценаріїв повсякденного використання. В результаті реалізовано систему, що підвищує комфорт проживання, знижує енергоспоживання та надає користувачу гнучкий інструмент для подальшої адаптації під власні потреби. Отже, розроблена інформаційна система є ефективним прикладом впровадження технологій автоматизації у побутовому середовищі та має потенціал для подальшого розвитку.

# ВИСНОВКИ

У процесі дослідження було визначено оптимальну архітектуру для системи «розумний будинок», що забезпечує її ефективність, надійність, масштабованість і можливість подальшого апаратного розширення через аналіз даних. Система спроектована таким чином, щоб задовольняти вимоги до інтеграції пристроїв, гнучкості налаштувань та високої безпеки даних, що є важливим аспектом для забезпечення стабільної роботи в реальному часі.

Завдяки аналізу сучасних тенденцій у сфері автоматизації житлових приміщень були виявлені основні проблеми, такі як відсутність єдиних стандартів, сумісність різних пристроїв, а також потреба в забезпеченні конфіденційності та захисту даних. Це дозволило сформулювати вимоги до технологій, необхідних для побудови «розумного будинку».

Для досягнення поставленої мети було обрано платформу Home Assistant, яка найбільш відповідає вимогам щодо функціональності, безпеки, гнучкості та масштабованості. Платформа дозволяє ефективно інтегрувати пристрої, забезпечує локальне виконання сценаріїв без використання сторонніх хмарних сервісів, що важливо для збереження конфіденційності даних та підтримки стабільності роботи системи.

Процес вибору програмних і апаратних компонентів базувався на детальному аналізі вимог до продуктивності, енергоефективності та сумісності. Обрані компоненти забезпечили належний рівень стабільності роботи системи, а також дозволили ефективно інтегрувати різноманітні сенсори та актуатори для моніторингу та управління функціями будинку.

Архітектура системи побудована на локальній клієнт-серверній моделі, що гарантує високий рівень швидкодії, масштабованості та автономності. Основні компоненти системи — збір, обробка, зберігання даних і управління пристроями — дозволяють гнучко адаптувати систему до змін в середовищі, забезпечуючи високий рівень автоматизації та зниження енергоспоживання.

Інтеграція модуля аналізу даних дозволяє системі автоматично реагувати на зміни навколишнього середовища, оптимізуючи роботу пристроїв у

реальному часі. Це підвищує комфорт проживання і дозволяє користувачам налаштовувати систему відповідно до своїх індивідуальних потреб.

Розроблена система є ефективним прикладом застосування технологій автоматизації в побутовому середовищі. Вона забезпечує високу стабільність роботи, економію енергії та підвищення рівня комфорту. Крім того, система демонструє значний потенціал для подальшого розвитку, включаючи розширення функціональності через інтеграцію нових технологій та пристроїв.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кришталь С.О., Проценко Д.П. Порівняння варіантів архітектури інформаційної системи “розумний будинок”. Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (Вінниця, 2025)». 2025. URL: https://conferences.vntu.edu.ua/index.php/mn/mn2025/author/submission/25244 (дата звернення: 28.05.2025).

2. Побудова інтеграційного рівня управління мережі IoT на базі протоколу ZigBee. 2021. URL: https://ela.kpi.ua/items/4e9330ca-7474-48eb-ba98-ca23f20e529e (дата звернення: 28.05.2025).

3. MQTT Version 3.1.1 Specification. OASIS Open. 2014. URL: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html (дата звернення: 28.05.2025).

4. Home Assistant Documentation. Home Assistant. 2023. URL: https://www.home-assistant.io/docs/ (дата звернення: 28.05.2025).

5. Smart Home Human Activity Simulation Tool for OpenHab-based Research. IoT Garage. 2020. URL: https://iotgarage.net/publications/pdfs/SmartHomeSimulationTool.pdf (дата звернення: 28.05.2025).

6. MQTT: The Standard for IoT Messaging. MQTT.org. 2023. URL: https://mqtt.org/ (дата звернення: 28.05.2025).

7. Smart Homes and Home Automation. Berg Insight AB. 2019. URL: http://www.berginsight.com/ReportPDF/ProductSheet/bi-sh7-ps.pdf (дата звернення: 28.05.2025).

8. Побудова інтеграційного рівня управління мережі IoT на базі протоколу ZigBee. 2021. URL: https://ela.kpi.ua/items/4e9330ca-7474-48eb-ba98-ca23f20e529e (дата звернення: 28.05.2025).

9. A Smart Home Automation technique with Raspberry Pi using IoT. IEEE. 2016. URL: https://ieeexplore.ieee.org/abstract/document/7873584 (дата звернення: 28.05.2025).

10. Smart home energy management system using IEEE 802.15.4 and ZigBee. IEEE. 2010. URL: https://ieeexplore.ieee.org/abstract/document/5606276 (дата звернення: 28.05.2025).

11. An open source and low-cost Smart Auditorium. Université de Liège. 2021. URL: https://orbi.uliege.be/handle/2268/263338 (дата звернення: 28.05.2025).

12. Поджаренко В.О., Кучерук В.Ю., Севастьянов В.М. Основи мікропроцесорної техніки. Навчальний посібник. Вінниця: ВНТУ, 2006. 226 с.

13. Schulze H.-J. Smart Home Automation with Linux and Raspberry Pi. Apress. 2021. 290 c.

14. YAML Based Input File Format For Finite Element Analysis. NTNU Open. 2020. URL: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2621457 (дата звернення: 28.05.2025).

15. Zaidan A.A., Zaidan B.B. A review on intelligent process for smart home applications based on IoT: coherent taxonomy, motivation, open challenges, and recommendations. Artificial Intelligence Review. 2020. Vol. 53. pp. 141–165. DOI: https://doi.org/10.1007/s10462-018-9648-9 (дата звернення: 28.05.2025).

16. A Study of PM2.5 and $CO_2$ Dynamics Using Low-Cost Sensors. MDPI. 2024. URL: https://www.mdpi.com/2076-3298/11/11/237 (дата звернення: 28.05.2025).

17. Using the ESP32 Microcontroller for Data Processing. IEEE. 2019. URL: https://ieeexplore.ieee.org/abstract/document/8765944 (дата звернення: 28.05.2025).

18. Інтерактивна автоматична система «Розумний Будинок». Державний університет телекомунікацій. 2023. URL: https://con.dut.edu.ua/index.php/communication/article/view/2401 (дата звернення: 28.05.2025).

19. Smart Home Human Activity Simulation Tool for OpenHab-based Research. IoT Garage. 2020. URL: https://iotgarage.net/publications/pdfs/SmartHomeSimulationTool.pdf (дата звернення: 28.05.2025).

20. Adaptive-Predictive Control Strategy for HVAC Systems in Smart Buildings – A Review. ResearchGate. 2020. URL: https://www.researchgate.net/publication/346782370_Adaptive-predictive_control_strategy_for_HVAC_systems_in_smart_buildings_-_A_review (дата звернення: 28.05.2025).

Додаток А

(обов’язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

“____”_____ 2025 р.

# ТЕХНІЧНЕ ЗАВДАННЯ

на комплексну бакалаврську кваліфікаційну роботу

«ІНФОРМАЦІЙНА СИСТЕМА «РОЗУМНИЙ БУДИНОК» НА БАЗІ ЛОКАЛЬНОЇ КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ». АНАЛІЗ ДАНИХ ТА ВИБІР ОПТИМАЛЬНОЇ АРХІТЕКТУРИ СИСТЕМИ

08-34.БКР.012.00.000 ТЗ

Керівник: к.т.н., доц.

_____ Дмитро ПРОЦЕНКО

«__» _____ 2025 р.

Розробив студент гр. 2ІСТ-21б

_____ Сергій КРИШТАЛЬ

«__» _____ 2025 р.

Вінниця 2025

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2025р., та індивідуальне завдання на БКР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2025р.

2. Джерела розробки:

1) Home Assistant Automating, інформаційний ресурс. URL: https://www.home-assistant.io/docs/automation/

2) Сервіс Home Assistant, інформаційний ресурс URL: https://www.home-assistant.io/.

3. Мета і призначення роботи.

Метою дослідження є вибір оптимальної архітектури з метою обґрунтування для забезпечення її ефективності, надійності та масштабованості, і подальшого апаратного розширення системи.

4. Вихідні дані для проведення робіт:

Платформа Home Assistant, встановлена на міні-комп'ютері, сенсорні пристрої (датчики температури, руху, вологості), протоколи ESPhome, Zigbee, MQTT, Wi-Fi, YAML-конфігурації та середовище автоматизації з підтримкою локальної обробки подій.

5. Методи дослідження:

Аналіз існуючих систем розумного дому, порівняння функціональних можливостей платформ, моделювання сценаріїв автоматизації, тестування інтеграції з пристроями.

6. Етапи роботи і терміни їх виконання:

a) Аналіз предметної області _____ – _____
b) Порівняльний аналіз існуючих систем _____ – _____
c) Вибір оптимальної архітектури та технологій _____ – _____
d) Розроблення системи «розумний будинок» _____ – _____
e) Оформлення матеріалів до захисту БКР _____ – _____

7. Очікувані результати та порядок реалізації

Отримання працездатної системи типу «розумний будинок» з локальним управлінням, можливістю інтеграції сенсорами та пристроями, підтримкою автоматизації подій і обробки даних.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»)».

9. Порядок приймання роботи

Публічний захист «___» _____ 2025 р.
Початок розробки «___» _____ 2025 р.
Граничні терміни виконання БКР «___» _____ 2025 р.

Розробив студент групи 2ІСТ-21б _____ Сергій КРИШТАЛЬ

Додаток Б

(обов’язковий)

**ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Назва роботи: «Інформаційна система «розумний будинок» на базі локальної клієнт-серверної архітектури. Аналіз даних та вибір оптимальної архітектури системи

Тип роботи: бакалаврська кваліфікаційна робота
Підрозділ: кафедра САІТ, ФІІТА, гр. 2ІСТ-21б

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 2.38%

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ■ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- □ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- □ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Віталій МОКІН, зав. каф. САІТ _____ (підпис)

Євгеній КРИЖАНОВСЬКИЙ, доц. каф. САІТ _____ (підпис)

Особа, відповідальна за перевірку _____ (підпис) Сергій ЖУКОВ

З висновком експертної комісії ознайомлений(-на)

Керівник _____ (підпис) Дмитро ПРОЦЕНКО, к.т.н., доц. каф. САІТ

Здобувач _____ (підпис) Сергій КРИШТАЛЬ

Додаток В

(обов’язковий)

**ІЛЮСТРАТИВНА ЧАСТИНА**

«ІНФОРМАЦІЙНА СИСТЕМА «РОЗУМНИЙ БУДИНОК» НА БАЗІ ЛОКАЛЬНОЇ КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ». АНАЛІЗ ДАНИХ ТА ВИБІР ОПТИМАЛЬНОЇ АРХІТЕКТУРИ СИСТЕМИ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2025 р.

| Критерій | 2-рівнева | 3-рівнева | багаторівнева |
|---|---|---|---|
| Швидкодія | + Найвища | +- Середня | - Нижча |
| Масштабованість | - Обмежена | +- Помірна | + Висока |
| Складність розгортання | + Проста | +- Середня | - Висока |
| Безпека | - Низька | +- Середня | + Висока |
| Вартість | + Низька | +- Середня | - Висока |

Рисунок В.1 – Таблиця порівняння архітектур


| Критерій | 2-рівнева | 3-рівнева | багаторівнева |
|---|---|---|---|
| Затримка | 50мс | 100-150мс | 200+мс |
| Безпека | Низька | Середня | Висока |
| Маштаб | 1 сервер | До 5 серверів | Необмежено |
| Складність | Просто | Помірно | Складно |

Рисунок В.2 – Вибір архітектури

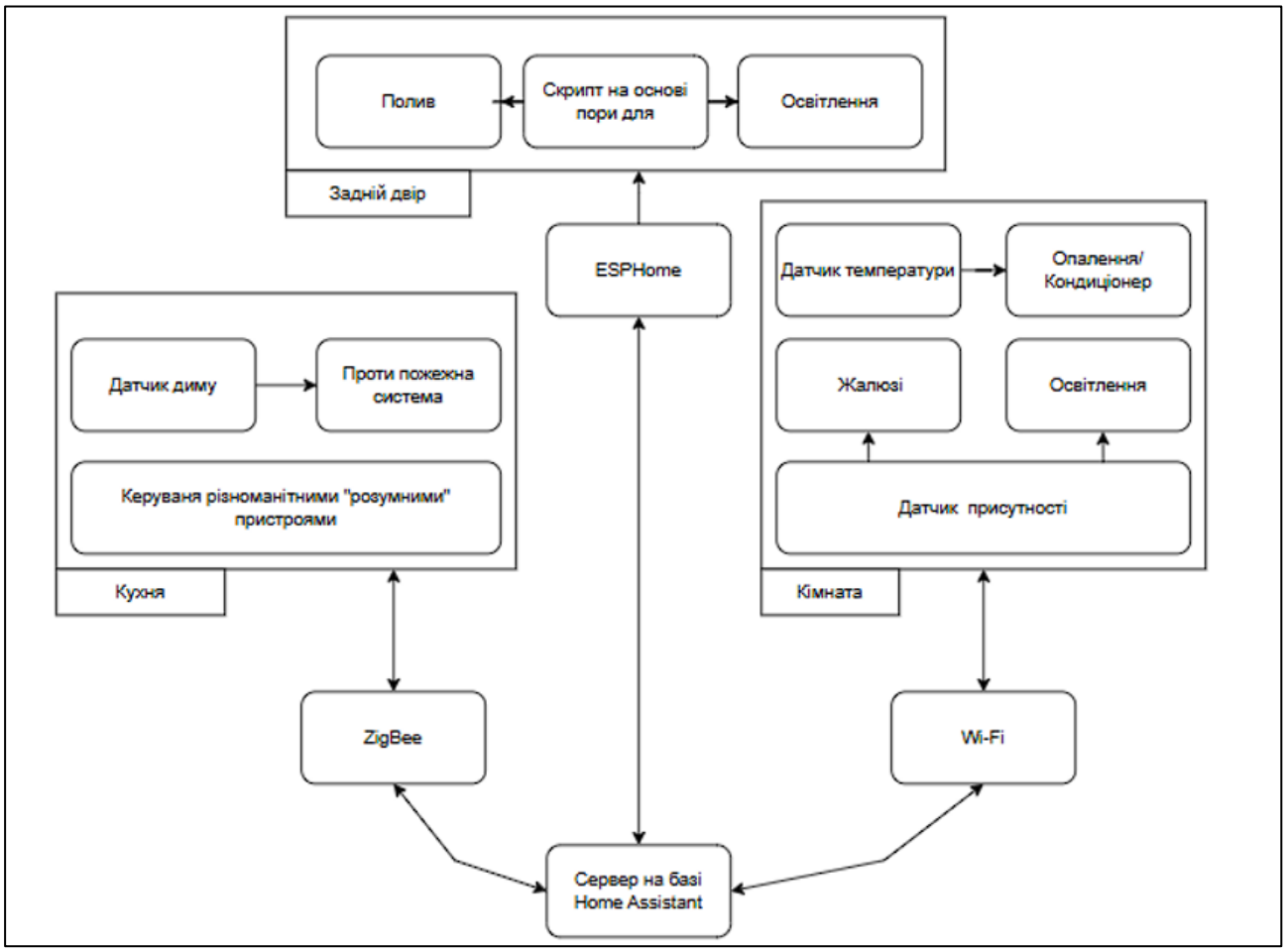


Рисунок В.3 – Структурна схема «розумного будинку»

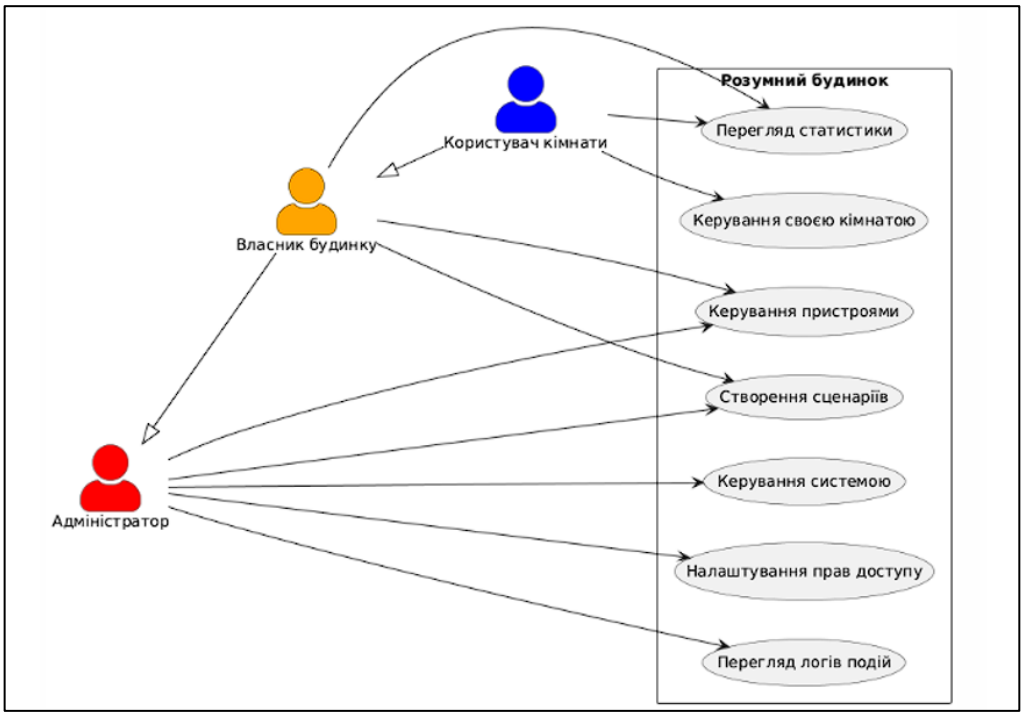


Рисунок В.4 – UML Use Case діаграма для системи "розумний будинок" з різними ролями користувачів

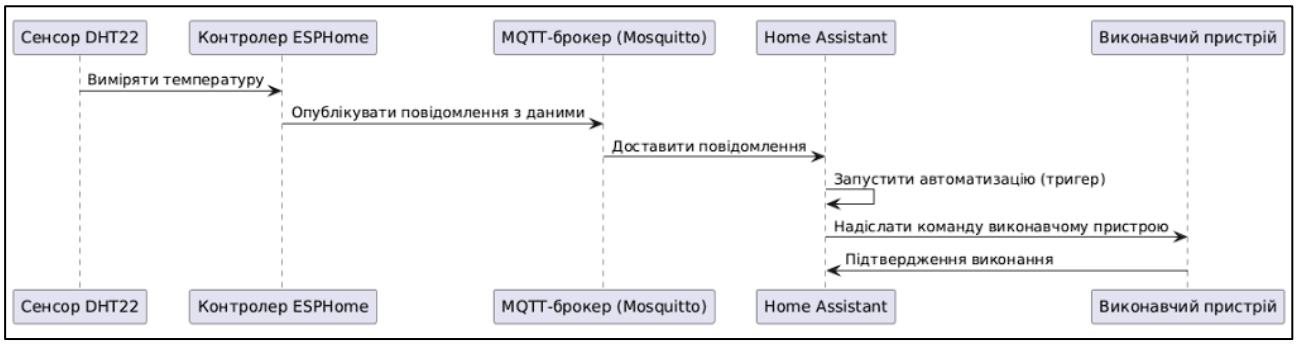


Рисунок В.5 – Діаграма послідовності взаємодії сенсора, MQTT-брокера, Home Assistant та виконавчого пристрою

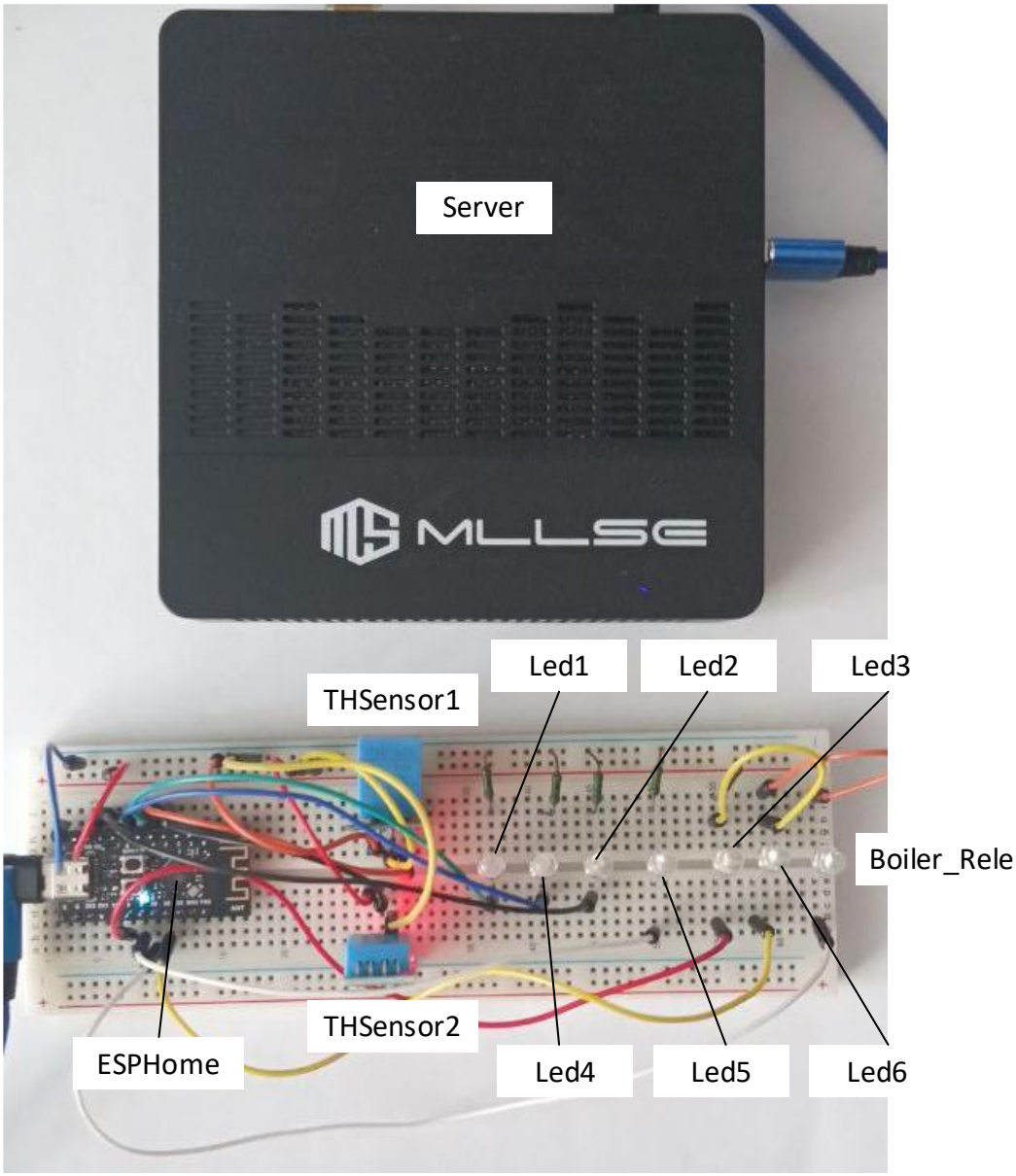


Рисунок В.6 - Апаратна реалізація інформаційної системи «розумний будинок» у вигляді макету